

NAVAL POSTGRADUATE SCHOOL

Monterey, California



CONTRACTOR REPORT

PROCEDURE AND COMPUTER PROGRAM
FOR THE APPROXIMATION OF DATA
(WITH APPLICATION TO MULTIPLE SENSOR PROBES)

H. ZEBNER
BDM CORPORATION
P.O. Box 2019
Monterey, CA 93940

August 1980

Interim Report for Period June 1980 - August 1980

Approved for public release; distribution unlimited

Prepared for:
Naval Postgraduate School
Monterey, California 93940

FedDocs
D 208.14/2
NPS-67-80-001CR

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral J. J. Ekelund
Superintendent

D. A. Schradny
Acting Provost

The work reported herein was carried out by the BDM Corporation under Work Order 426, Turbopropulsion Laboratory Support, under Contract Number N00014-78-C-0204. The work was part of a program entitled Transonic Compressor Investigations, funded in part by Naval Air Systems Command under Program Element 61153N, Contract Number N00019-80-WR-01199. The program was under the cognizance of Dr. H. J. Mueller, Code 310.

The author is a candidate for the degree D. Ing under Prof. H. Gallus at Rhein,-Westf. Techn. Hochschule, 51 Aachen, W. Germany.

This report was prepared by:

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS67-80-001CR	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PROCEDURE AND COMPUTER PROGRAM FOR THE APPROXIMATION OF DATA (WITH APPLICATION TO MULTIPLE SENSOR PROBES).		5. TYPE OF REPORT & PERIOD COVERED CONTRACTOR REPORT June 1980 - August 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) H. Zebner		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0204
9. PERFORMING ORGANIZATION NAME AND ADDRESS BDM CORPORATION P.O. BOX 2019 MONTEREY, CALIFORNIA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Order 426
11. CONTROLLING OFFICE NAME AND ADDRESS NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA 93940		12. REPORT DATE August 1980
		13. NUMBER OF PAGES 186
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Interim Report. Principal Investigator, Dr. R. P. Shreeve, Director Turbopropulsion Laboratory, Naval Postgraduate School, Monterey, California 93940		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Data Approximation Probe Calibration Flow Measurements		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A procedure to approximate data given at arbitrary intervals in two-, three- and four dimensions using polynomial expressions is described. The programming of the problem is explained in each case and a user manual is given for software implemented on the TPL Hewlett-Packard computer system. The method, which is general, was derived and has been applied to represent the calibration of flow probes which have multiple sensors.		

TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE NO.</u>
ABSTRACT	3
1. INTRODUCTION	4
2. TWO DIMENSIONAL APPROXIMATION	5
2.1. Problem	5
2.2. Approach	6
2.3. Solution	7
2.4. Equation System	9
2.4.1. System Matrix A	9
2.4.2. Right Hand Side Vector B	10
2.5. Software	12
2.6. Sample User Program	15
2.6.1. Listing	15
2.6.2. Load Map	18
2.6.3. Results	19
3. THREE DIMENSIONAL APPROXIMATION	21
3.1. Problem	21
3.2. Approach	22
3.3. Solution	24
3.4. Equation System	27
3.4.1. System Matrix A	27
3.4.2. Right Hand Side Vector B	33
3.5. Software	35
3.6. Sample User Program	37
3.6.1. Listing	37
3.6.2. Load Map	43
3.6.3. Results	44
4. FOUR DIMENSIONAL APPROXIMATION	49
4.1. Problem	49
4.2. Approach	50
4.3. Solution	53
4.4. Structure of Equation System	67
4.4.1. System Matrix A	67
4.4.2. Right Hand Side Vector B	76
4.5. Software	79
4.6. Sample User Program	81
4.6.1. Listing	81
4.6.2. Load Map	83
4.6.3. Results	84

5.	CONCLUSIONS AND RECOMMENDATIONS	87
6.	LIST OF REFERENCES	88
Appendix A	SOME USEFUL MATRIX CONVENTIONS AND OPERATIONS	89
	A1 Submatrix Notation	89
	A2 Diagonal Lines and Bands 1. Order	89
	A3 Diagonal Lines and Bands 2. Order	90
Appendix B	SOFTWARE DESCRIPTION: FLOW CHARTS	92
Appendix C	SOFTWARE DESCRIPTION: LISTINGS	130
	DISTRIBUTION LIST	180

ABSTRACT

A procedure to approximate data given at arbitrary intervals in two-, three- and four dimensions using polynomial expressions is described. The programming of the problem is explained in each case and a user manual is given for software implemented on the TPL Hewlett-Packard computer system. The method, which is general, was derived and has been applied to represent the calibration of flow probes which have multiple sensors.

1. INTRODUCTION

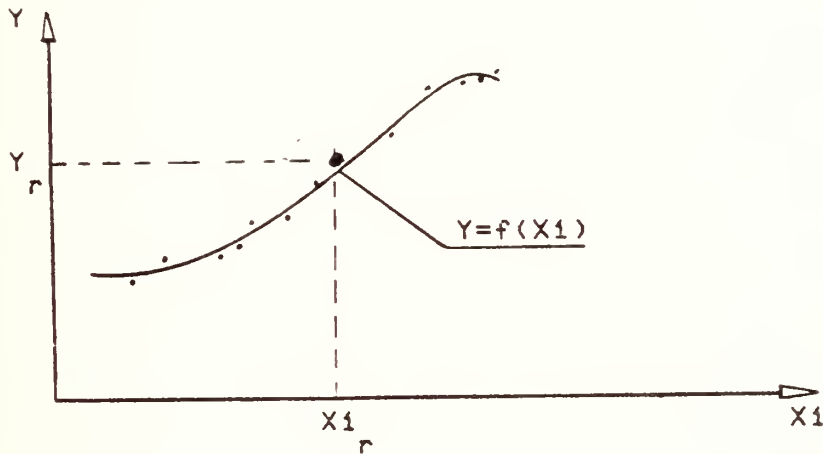
This report describes algorithms to approximate data patterns such as those which occur, for instance, when pneumatic-velocity probes are calibrated. The functional value (Y) can depend on either one (X_1), two (X_1 and X_2) or even three (X_1 , X_2 and X_3) parameters. The approximation $Y = f(X_1)$ will be referred to as the two dimensional, the approximation $Y = f(X_1, X_2)$ as the three dimensional and the approximation $Y = f(X_1, X_2, X_3)$ as the four dimensional approximation. These three options meet the requirements for probes used in the Turbopropulsion Laboratory.

Using the least-squares criterion to obtain the coefficients in assumed polynomials leads to a system of linear equations, which are, in principle easily solved. Numerical problems may arise however, since the software, as described herein and as implemented in the NPS Turbopropulsion Laboratory Hewlett Packard 21MX computer, uses 32-bit real constants.

The author wishes to express his thanks to Professor Ray Shreeve, whose constant and critical interest was a crucial help to solve this mathematical problem and link it to engineering application.

2. TWO DIMENSIONAL APPROXIMATION

2.1. Problem:



A data set of NPNTS1 data points is given, where Y depends on parameter X_1 . The data pattern is to be approximated by a function $Y = f(X_1)$, so that the error between data points and analytically determined points is lowest.

2.2. Approach:

Polynominals are commonly used in order to approximate data patterns. $Y = f(X_1)$ is a function which approximates the data value, V , at each value of the variable X_1 , we look for our expression of the form

$$Y = C_1 + C_2 * X_1 + C_3 * X_1^2 + \dots + C_L * X_1^{(L-1)}$$

or

$$Y = \sum_{i=1}^L C_i * X_1^{(i-1)} \quad (2.1.)$$

in which the coefficients C_i are to be determined by the method of least squares. As Ref. 1 shows the least squares criterion leads to a linear equation system that can easily be solved. We define the error

$$R = \sum_{r=1}^{NPNTS1} [f(X_{1r}) - Y_r]^2 \quad (2.2.)$$

where the index r denotes the individual data point. Using equation (2.1.) R becomes

$$R = \sum_{r=1}^{NPNTS1} [C_1 + C_2 * X_{1r} + C_3 * X_{1r}^2 + \dots + C_L * X_{1r}^{(L-1)} - Y_r]^2$$

2.3. Solution:

R depends on the selection of the coefficients C_1, \dots, C_L . In order to minimize the error, R is differentiated with respect to C_1, \dots, C_L and the partial derivatives are set to zero. Thus

$$\frac{\partial R}{\partial C_i} = 0 \quad i = 1, \dots, L$$

or

$$\frac{\partial R}{\partial C_i} = \sum_{r=1}^{NPNTS1} 2 * [C_1 + C_2 * X_{1r} + C_3 * X_{1r}^2 + \dots + C_L * X_{1r}^{(L-1)} - Y_r] * \frac{\partial [C_1 + C_2 * X_{1r} + C_3 * X_{1r}^2 + \dots + C_L * X_{1r}^{(L-1)} - Y_r]}{\partial C_i} = 0$$

Assuming that the summations extend over all the data points, \sum should be understood to mean $\sum_{r=1}^{NPNTS1}$. Performing the differentiations and rearranging the equations, we get

$$\begin{aligned} NPNTS1 * C_1 + \sum_r X_{1r} * C_2 + \sum_r X_{1r}^2 * C_3 + \dots + \sum_r X_{1r}^{(L-1)} * C_L &= \sum_r Y_r \\ \sum_r X_{1r} * C_1 + \sum_r X_{1r}^2 * C_2 + \sum_r X_{1r}^3 * C_3 + \dots + \sum_r X_{1r}^L * C_L &= \sum_r Y_r * X_{1r} \\ \sum_r X_{1r}^2 * C_1 + \sum_r X_{1r}^3 * C_2 + \sum_r X_{1r}^4 * C_3 + \dots + \sum_r X_{1r}^{(L+1)} * C_L &= \sum_r Y_r * X_{1r}^2 \\ \dots\dots\dots \\ \sum_r X_{1r}^{(L-1)} * C_1 + \sum_r X_{1r}^L * C_2 + \sum_r X_{1r}^{(L+1)} * C_3 + \dots + \sum_r X_{1r}^{(2L-2)} * C_L &= \sum_r Y_r * X_{1r}^{L-1} \end{aligned}$$

In matrix notation

$$\begin{pmatrix}
 \text{NPNTS1} & \sum x_{1r} & \sum x_{1r}^2 & \dots & \sum x_{1r}^{L-1} \\
 \sum x_{1r} & \sum x_{1r}^2 & \sum x_{1r}^3 & \dots & \sum x_{1r}^L \\
 \sum x_{1r}^2 & \sum x_{1r}^3 & \sum x_{1r}^4 & \dots & \sum x_{1r}^{L+1} \\
 \dots & \dots & \dots & \dots & \dots \\
 \sum x_{1r}^{L-1} & \sum x_{1r}^L & \sum x_{1r}^{L+1} & \dots & \sum x_{1r}^{2L-2}
 \end{pmatrix}
 \begin{pmatrix}
 c_1 \\
 c_2 \\
 c_3 \\
 \vdots \\
 c_L
 \end{pmatrix}
 =
 \begin{pmatrix}
 \sum Y_r \\
 \sum Y_r * x_{1r} \\
 \sum Y_r * x_{1r}^2 \\
 \vdots \\
 \sum Y_r * x_{1r}^{L-1}
 \end{pmatrix}
 \quad (2.3.)$$

- \hat{A} ... System Matrix
 \hat{C} ... Coefficients vector
 \hat{X} ... Right hand side vector

2.4. Equation System

2.4.1. System Matrix A

Since all Matrix elements on diagonal lines running up from left to right (2. order, as defined in Appendix A) are identical, the elements are renamed for simplification as follows:

$$\begin{aligned}
 a_1 &= a_y \\
 a_2 &= a_{21} = a_{12} \\
 a_3 &= a_{31} = a_{22} = a_{13} \\
 &\vdots \\
 a_L &= a_{L1} = a_{L-1,2} = a_{L-2,3} = \dots = a_{2,L-1} = a_{1,L} \\
 a_{L+1} &= a_{L,2} = a_{L-1,3} = \dots = a_{3,L-1} = a_{2,L} \\
 a_{L+2} &= a_{L,3} = \dots = a_{4,L-1} = a_{3,L} \\
 &\vdots \\
 a_{2L-1} &= a_{L,L}
 \end{aligned}$$

so that the set of elements, a_k , can be written as

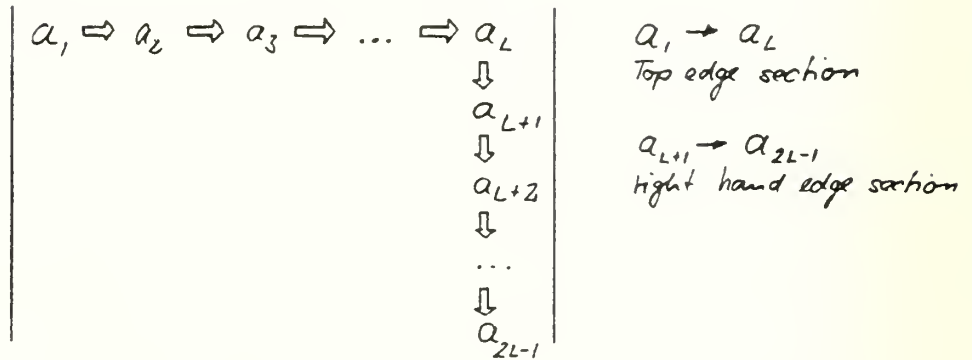
$$a_k = \sum_{r=1}^{NPNTS1} X1_r^{k-1} \quad k=1, \dots, 2L-1 \quad (2.4.)$$

This operation is programmed in REAL FUNCTION S2 (NPNTS1,IPOWR1,IY).

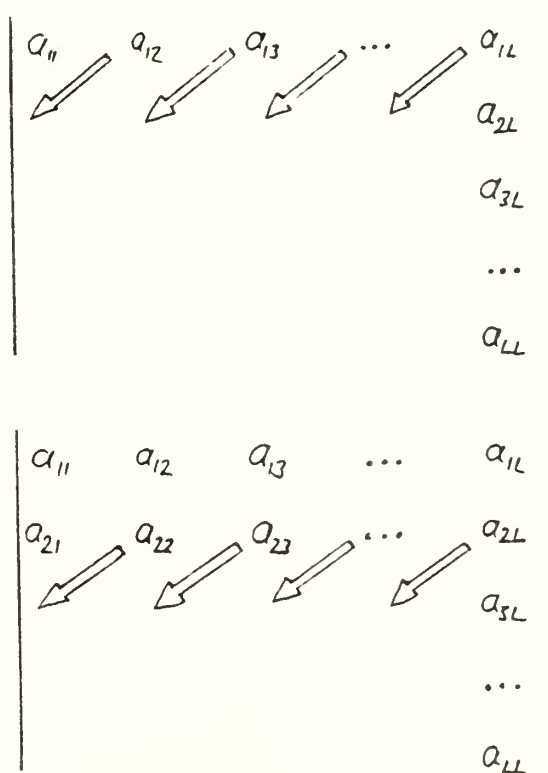
The data X1 and Y are known to this function through a common block named DTA2. So only NPNTS1, IPOWR1 (= k-1) and IY (=0) have to be passed to the function, and the value of a_k is returned through the function name S2.

SUBROUTINE MAT2 presets the system matrix in the following way:

i) Preset edge section elements (using REAL FUNCTION S2)



ii) Copy defined elements diagonally



$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1L} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2L} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3L} \\ \dots & \dots & \dots & \dots & \dots \\ a_{L1} & a_{L2} & a_{L3} & \dots & a_{LL} \end{pmatrix}$$

2.4.2. Right hand side vector B

The elements of the right hand side vector B can be written as

$$b_k = \sum_{r=1}^{NPNTS1} (Y_r \cdot X1_r^{k-1}) \quad k = 1, \dots, L \quad (2.5.)$$

To calculate b_k , again REAL FUNCTION S2 is used. Data X1 and Y are available through COMMON block DTA2. NPNTS1, IPOWR1 (=k-1) and IY (=1) have to be passed to the function, and the value of b_k is returned through the function name S2.

2.5. Software:

The software to compute the coefficients for a two dimensional approximation is described in Appendix B and is implemented in the TPL HP-21MX computer system

To work correctly with these program modules, the user must conform to the following conventions:

- i) Provide the data in two arrays (Type: REAL) of 256 elements through a COMMON block, named DTA2.

```
COMMON / DTA2 / X1,Y
REAL X1(256),Y(256)
```

- ii) Dimension an array (Type: REAL) of 7 (seven) elements to contain the coefficients.

```
REAL COEF(7)
```

- iii) Define the parameters NPNTS1, L and IPRINT (Type: all INTEGER)

```
NPNTS1    ...  # of data points
            L ≤ NPNTS1 ≤ 256
```

```
L          ...  (desired order of polynominal) + 1
            1 ≤ L ≤ 7
```

```
IPRINT     ...  controls quantity of print out
2          ...  Print system matrix and right
                hand side vector before and
                after Gauss Jordan Elimination
1          ...  Print equation system after
                Gauss Jordan Elimination
<0         ...  No print out
>0         ...  Print equation (1,1) with the
                actual parameters
```

- iv) When loading a program, that uses the subroutine MAT2, the binary library file has to be searched for externals.

Suppose, the source file &USER::26 contains a user program named USER. This program calls MAT2. After the compilation, the relocatable binary file for this user program is %USER::26. To load the program, the following procedure is recommended:

Type (from FMGR)

:RU,LOADR

and the loader program will respond

/LOADR: RE,%USER::26

where the underlined information already is the user's input. This causes the loader to load all program modules of the user program; a load map is listed on the terminal. Upon completion the loader prompts

/LOADR: MS,%TPLBL::26

where the underlined information already is the user's input. Now the loader conducts a search for all unsatisfied externals of the user program in the binary library file %TPLBL::26. Since some library programs have externals themselves the search has to be repeated (MS ... multiple search) as many time as is necessary for all externals to be satisfied. The loader prompts

/LOADR: END

where the underlined information is the user's input. The loader now loads all system programs, outputs a load map and generates the program. Upon completion, the loader outputs a ready message

/LOADR:USER READY AT 1:28 PM WED., 10 SEPT, 1980
/LOADR: END

Now the program can be run

:RU,USER

Again, the underlined information is the user's input.

If all these requirements are met, the correct call for the subroutine is:

CALL MAT2 (NPNTS1,L,COEF,IPRINT)

Upon completion, the array COEF contains the coefficients (COEF(1)=C₁, ..., COEF(L)=C_L). Externals used by MAT2 are: AB2, DTA2, S2 under no circumstances may the user use any of these names for modules of

his own user program. In some cases the program may not be able to perform a Gauss Jordan Elimination to the system matrix and the right hand side vector and thus cause the program to stop. If this happens, an error message is displayed.

It is highly encouraged, to use the system function FNP to calculate the value of a polynomial at one specified X1 rather than a loop such as the following:

```
S=.0
DO 08 I=1,L,1
08 S=S+X**(I-1)*COEF(I)
YCALC=S
```

To call FNP is easier and faster, because FNP is a programmed Horner Scheme. The recommended call is then;

```
L1=L-1
YCALC=FNP(COEF,X,L1)
```

2.6. Sample User Program

2.6.1. Listing

PAGE 0001 FTN. 12:40 PM THU., 18 SEP., 1980

```

0001 FTN4,L
0002 PROGRAM DEMO2 (3,99)
0003 C .....
0004 C .....
0005 C THIS IS A DEMONSTATION PROGRAM AND IT SHOWS THE CORRECT USE
0006 C OF THE TPL BINARY LIBRARY (ON TYPE 6 FILE XTPLBL) USING
0007 C EXAMPLES. FOR FURTHER QUESTIONS READ THE TPL-LIBRARY-BINDER
0008 C OR CONSULT THE SOFTWARE MANAGER .
0009 C .....
0010 C .....
0011 C * DEMONSTRATE THE USE OF THE TPL BINARY LIBRARY.
0012 C .....
0013 COMMON / AFLD / A
0014 COMMON / AFLD / A
0015
0016 REAL A(256)
0017 REAL A(256), Y(256)
0018
0019 COMMON / AFLD / A
0020 INTEGER NOLF, NOCR(2), ICLR(3)
0021
0022 DATA NOLF /006537B/
0023 DATA NOCR /000033B,040433B/
0024 DATA ICLR /015524B,015515B,006537B/
0025 DATA PI /3.141593/
0026
0027 C FORMATS DEMO2 START
0028 101 FORMAT (///" HELLO! THIS IS PROGRAM DEMO AND WE'LL SEE H
0029 *OW TO USE THE MARVELLOUS TPL BINARY" A2/" LIBRARY! YOUR INTEREST
0030 *WILL BE GREATLY AWARDED BY EASIER PROGRAMMING."///)
0031 102 FORMAT (" GENERATING DATA POINTS"/5X"I"5X"X(I)"5X"Y(I)" )
0032 103 FORMAT (2X,14,2F9.3,A2)
0033 104 FORMAT (" DATA POINTS GENERATED!"/)
0034 105 FORMAT (" INITIALIZING THE PLOTTER")
0035 106 FORMAT (" PLOTTER INITIALIZED!"/)
0036 107 FORMAT (" DEFINE PLOTTER AND USER AREAS AND DRAW AXES")
0037 108 FORMAT (9X"6X"XMIN "6X"XMAX "6X"YMIN
0038 *"6X"YMAX"/9X,4("10X" ")/
0039 *" HP9872",4(" F10.3""),"/
0040 *" USER ",4(" F10.3""),"/)
0041 109 FORMAT (" PLOTTER AREAS DEFINED AND AXES DRAWN!"/)
0042 110 FORMAT (" DRAWING DATA POINTS INTO COORDINATES SYSTEM")
0043 111 FORMAT (" DATA POINTS DRAWN!"/)
0044 112 FORMAT (" CALCULATING A CURVE FIT THROUGH THE DATA POINTS
0045 *")
0046 113 FORMAT (" ENTER DEGREE OF POLYNOMIAL (NORDER) TO FIT THRO
0047 *UGH THE POINTS "2A2)
0048 114 FORMAT (" ENTER IPRINT "2A2)
0049 115 FORMAT (/ " CURVE FIT DONE!"/)
0050 116 FORMAT (" CURVE FIT DONE!"/)
0051 117 FORMAT (" PLOTTING THE CURVE FIT")
0052 118 FORMAT (" CURVE FIT DRAWN!"/)
0053 149 FORMAT ("((3A2)))
0054 C FORMATS DEMO2 STOP
0055
0056 C .....
0057 C .....
0058 C .....
0059 C GET THE LU OF THE TERMINAL.
0060 C .....
0061 C .....
0062 C LI = LOGLU(I)
0063 C WRITE (LI, 101) NOLF
0064 C .....
0065 C .....
0066 C .....
0067 C .....
0068 C .....
0069 C GENERATE DATA POINTS SCATTERED AROUND A POLYNOMIAL.
0070 C .....
0071 C .....
0072 C CALL INITG (13)
0073 C WRITE (LI, 102)
0074 C XSTART = -6.
0075 C XSTOP = +2.

```

```

0076      .....
0077      DX = (XSTOP-XSTART)/(NPNTS1-1)
0078      NORDER = 4
0079      COEF(1) = -1.
0080      COEF(2) = -0.500000
0081      COEF(3) = -0.000340
0082      COEF(4) = +0.300000
0083      COEF(5) = +0.055000
0084      A = (2.0*PI)*0.03125
0085      DO 01 I=1, NPNTS1-1
0086      X1(I) = XSTART+(I-1)*DX
0087      Y(I) = FNP(COEF,X1(I),NORDER) + 0.25*SIN(X1(I)/A)
0088      01 WRITE (LI, 103) I,X1(I),Y(I),NOLF
0089      ISTOP1 = NPNTS1-1
0090      DO 02 I=1, ISTOP1,2
0091      DUMMY = Y(I+1)
0092      Y(I+1) = Y(I)
0093      02 Y(I) = DUMMY
0094      WRITE (LI, 149) (ICLR,I=1,3,1)
0095      WRITE (LI, 104)
0096
0097
0098
0099
0100      C C C C C
0101      : INITIALIZE THE PLOTTER.
0102      :
0103      :
0104      WRITE (LI, 105)
0105      LP = 13
0106      CALL INITG (13)
0107      WRITE (LI, 149) ICLR
0108      WRITE (LI, 106)
0109
0110
0111      C C C C C
0112      :
0113      : DEFINE PLOTTER AND USER AREAS; DRAW AXES.
0114      :
0115      :
0116      WRITE (LI, 107)
0117      XPMIN = 2.
0118      XPMAX = 12.
0119      YPMIN = 12.
0120      YPMAX = 22.
0121      XUMIN = -6.
0122      XUMAX = +2.
0123      YUMIN = -5.
0124      YUMAX = 10.
0125      WRITE (LI, 108) XPMIN,XPMAX,YPMIN,YPMAX,XUMIN,XUMAX,YUMIN,YUMAX
0126      ALPHAX = 0.00
0127      XA = (XPMAX - XPMIN)/(XUMAX-XUMIN)
0128      XB = (XPMIN*XUMAX-XPMAX*XUMIN)/(XUMAX-XUMIN)
0129      XL = (XPMAX - XPMIN)
0130      ALPHAY = 90.00
0131      YA = (YPMAX - YPMIN)/(YUMAX-YUMIN)
0132      YB = (YPMIN*YUMAX-YPMAX*YUMIN)/(YUMAX-YUMIN)
0133      YL = (YPMAX - YPMIN)
0134      CALL SETSM (113,1.)
0135      CALL AXIS (XPMIN,YPMIN,XL,ALPHAX,2HX1,+2,XUMIN,XUMAX,4HF4.1,4
0136      CALL AXIS (XPMIN,YPMIN,YL,ALPHAY,2HY,-2,YUMIN,YUMAX,2HI2,2,3
0137      WRITE (LI, 149) (ICLR,I=1,6,1)
0138      WRITE (LI, 109)
0139
0140
0141
0142      C C C C C
0143      :
0144      : PLOT DATA POINTS.
0145      :
0146      :
0147      WRITE (LI, 110)
0148      CALL SETSM (113,3.)
0149      DO 03 I=1, NPNTS1,1
0150

```



```

0151      XPLOT = X1(I)*XA+XB
0152      YPLOT = Y(I)*YA+YB
0153      CALL PLOT (XPLOT,YPLOT,2)
0154 03 CALL SYMBL (1)
0155      WRITE (LI, 149) ICLR
0156      WRITE (LI, 111)
0157
0158
0159
0160      C
0161      C
0162      C
0163      C
0164      C
0165      .....
0166      : CALCULATE CURVE FIT THROUGH DATA POINTS.
0167      :
0168      .....
0169      WRITE (LI, 112)
0170 04 WRITE (LI, 113) NOCR
0171      READ (LI, *) NORDER
0172      WRITE (LI, 149) ICLR
0173      IF ( NORDER .LT. 0 .OR. NORDER .GT. 6 ) GO TO 04
0174      WRITE (LI, 114) NOCR
0175      WRITE (LI, 115) IPRT
0176      WRITE (LI, 149) (ICLR,I=1,2,1)
0177      WRITE (LI, 116) IPRT
0178      IF (IPRINT.GE.0) WRITE (LI, 115)
0179      IF (IPRINT.LT.0) WRITE (LI, 116)
0180
0181      C
0182      C
0183      C
0184      C
0185      .....
0186      : PLOT POLYNOMIAL FIT.
0187      :
0188      .....
0189      WRITE (LI, 117)
0190      CALL SETSM (113,2.)
0191      DO 09 I=1,NPNTS1,1
0192      XPLOT = X1(I)*XA+XB
0193      YPLOT = FNP(COEF,X1(I),NORDER)*YA+YB
0194      IF (I.EQ.1) CALL PLOT (XPLOT,YPLOT,1)
0195 09 IF (I.GT.1) CALL PLOT (XPLOT,YPLOT,3)
0196      CALL SETSM (115,DUMMY)
0197      WRITE (LI, 149) ICLR
0198      WRITE (LI, 118)
0199
0200      C
0201      C
0202      C
0203      .....
0204      : TERMINATE GRAPHICS.
0205      :
0206      .....
0207      CALL STOPG
0208      STOP 7777
0209      END

```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01409 COMMON = 00000

2.6.2. Load map

DEMO2 10042 12642
FNP 12643 12746

DEMONSTRATE THE USE OF THE TPL BINARY LIBRARY
Calculate n-th order polyn

to load these programs, type (from LOADR): MS,%TPLBL

DTA2	12747	14746	2D-Approximation	/ DTA2	/	(DATA2 DTA2)
DATA2	14747	16777	2D-Approximation	arrange system matrix and vector		
AB2	17000	17131	2D-Approximation			compute summations
AB2	17132	17311	2D-Approximation	/ AB2	/	

AFLD	17312	20311	GSP	/ AFLD	/ A(256)	
AXIS	20312	22631	GSP			Draw and label axes
INITG	22632	23644	GSP	read control array A;		initialize plotter
PLOT	23645	24375	GSP			move pen to a defined point
SETSM	24376	25336	GSP			change plotter modes
STOPG	25337	25401	GSP			terminate graphics
SYMBL	25402	26136	GSP			print symbol at location of pen
FCTR	26137	26142	GSP	/ FCTR	/ FX,FY	
FACTR	26143	26162	GSP			vary size of the plot

LOGLU	26163	26240	92067-16268	REV.1903	790228
READF	26241	27202	92067-16125	REV.1940	790719
OPEN	27203	27500	92067-16125	REV.1903	790215
CLOSE	27501	27710	92067-16125	REV.1903	781229
CLRIO	27711	27717	750701	24998-16001	
OVRD.	27720	27720	92067-16125	REV.1903	780526
\$SMVE	27721	30007	92067-16268	REV.1903	790202
LURQ	30010	30372	92067-16268	REV.1903	790223
.DADS	30373	30502	780818	24998-16001	
.DMP	30503	30650	780818	24998-16001	
.DDI	30651	31151	781021	24998-16001	
SESSN	31152	31167	92067-16125	REV.1903	780413
R/W\$	31170	31326	92067-16125	REV.1903	781214
P.PAS	31327	31355	92067-16125	REV.1903	740801
.DNG	31356	31365	780818	24998-16001	
PAUSE	31366	31466	771122	24998-16001	
\$ALRN	31467	31604	92067-16268	REV.1903	770715
FMTIO	31605	33103	24998-16002	REV.1926	790417
ERRO	33104	33173	771122	24998-16001	
TAN	33174	33300	780424	24998-16001	
ABS	33301	33307	750701	24998-16001	
.SNCS	33310	33451	780424	24998-16001	
.DDE	33452	33463	780818	24998-16001	
.DIN	33464	33471	780818	24998-16001	
.RTOI	33472	33565	780921	24998-16001	
.FPWR	33566	33627	781106	24998-16001	
.SRT	33630	33670	770518	24998-16001	
.FCM	33671	33705	750701	24998-16001	
PAU.E	33706	33706	750701	24998-16001	
ERO.E	33707	33707	750701	24998-16001	
.CMRS	33710	33773	780424	24998-16001	
\$OPEN	33774	34150	92067-16125	REV.1903	790103
RW\$UB	34151	34516	92067-16125	REV.1903	781003
RWND\$	34517	34641	92067-16125	REV.1903	780801
FRMTR	34642	40277	24998-16002	REV.1926	790503
FMT.E	40300	40300	24998-16002	REV.1901	781107
IB4A2	40301	41601	59310-1X013	REV.1940	790802 1153
REIO	41602	41726	92067-16268	REV.1903	790316
RMPAR	41727	41771	781106	24998-16001	
PNAME	41772	42037	771121	24998-16001	
LUTRU	42040	42146	92067-16268	REV.1903	790223
IPUT	42147	42167	92067-16125	REV.1903	740801
\$SETP	42170	42214	781106	24998-16001	
.CFER	42215	42272	750701	24998-16001	
.LBT	42273	42323	770518	24998-16001	

15 PAGES RELOCATED 15 PAGES REQ'D NO PAGES EMA NO PAGES MSE
LINKS:BP PROGRAM:LB LOAD:TE COMMON:NC
/LOADR:DEMO2 READY AT 12:44 PM THU., 18 SEPT, 1980

/LOADR:\$END

2.6.3. Results

Affect of IPRINT on quantity of print out

i) printed output

System matrix A and vector B before Gauss Jordan Elimination

	¹	²	³	⁴	⁵	
1	.510E+02	-.102E+03	.487E+03	-.211E+04	.104E+05	-.578E+02
2	-.102E+03	.487E+03	-.211E+04	.104E+05	-.525E+05	.998E+02
3	.487E+03	-.211E+04	.104E+05	-.525E+05	.274E+06	-.113E+03
4	-.211E+04	.104E+05	-.525E+05	.274E+06	-.146E+07	.944E+03
5	.104E+05	-.525E+05	.274E+06	-.146E+07	.787E+07	.113E+05
	₁	₂	₃	₄	₅	

Equation system after Gauss Jordan Elimination

	¹	²	³	⁴	⁵	
1	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	-.970E+00
2	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	-.416E+00
3	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	-.162E-01
4	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.275E+00
5	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.514E-01
	₁	₂	₃	₄	₅	

IPRINT = 2

Coefficients COEF(I)

I = 1 -.970E+00
 I = 2 -.416E+00
 I = 3 -.162E-01
 I = 4 .275E+00
 I = 5 .514E-01

Equation system after Gauss Jordan Elimination

	¹	²	³	⁴	⁵	
1	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00	-.970E+00
2	.000E+00	.100E+01	.000E+00	.000E+00	.000E+00	-.416E+00
3	.000E+00	.000E+00	.100E+01	.000E+00	.000E+00	-.162E-01
4	.000E+00	.000E+00	.000E+00	.100E+01	.000E+00	.275E+00
5	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01	.514E-01
	₁	₂	₃	₄	₅	

IPRINT = 1

Coefficients COEF(I)

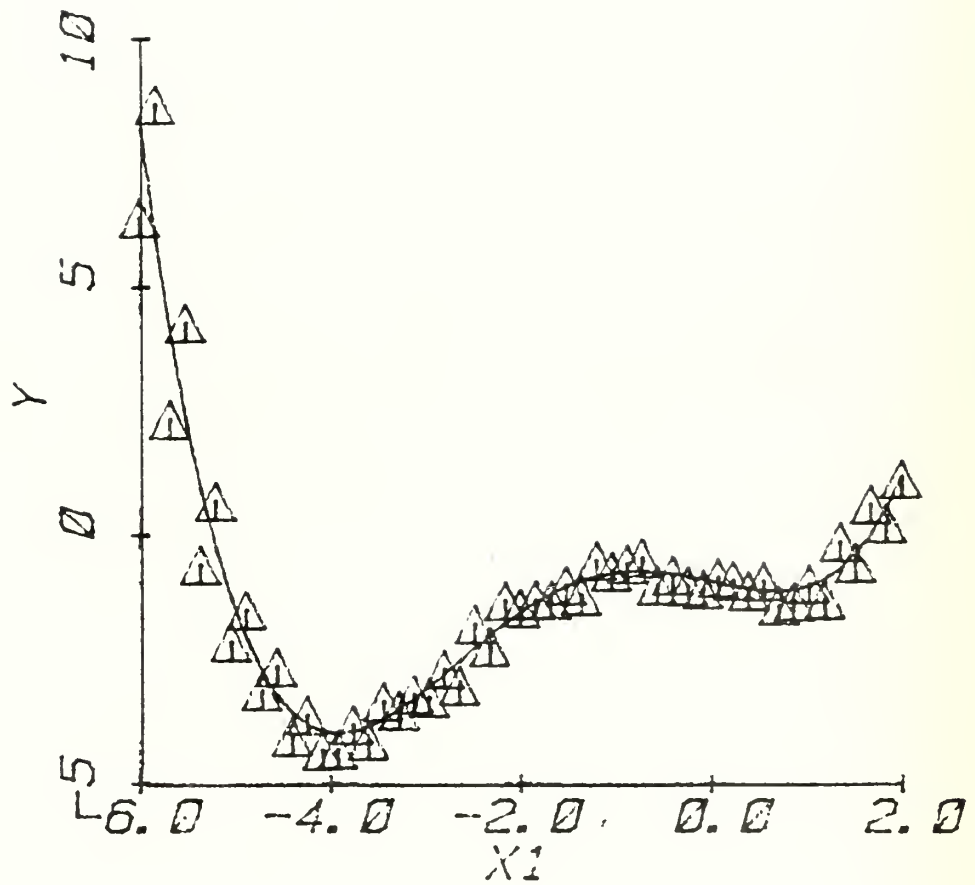
I = 1 -.970E+00
 I = 2 -.416E+00
 I = 3 -.162E-01
 I = 4 .275E+00
 I = 5 .514E-01

Coefficients COEF(I)

I = 1 -.970E+00
 I = 2 -.416E+00
 I = 3 -.162E-01
 I = 4 .275E+00
 I = 5 .514E-01

IPRINT < 1
 > 2

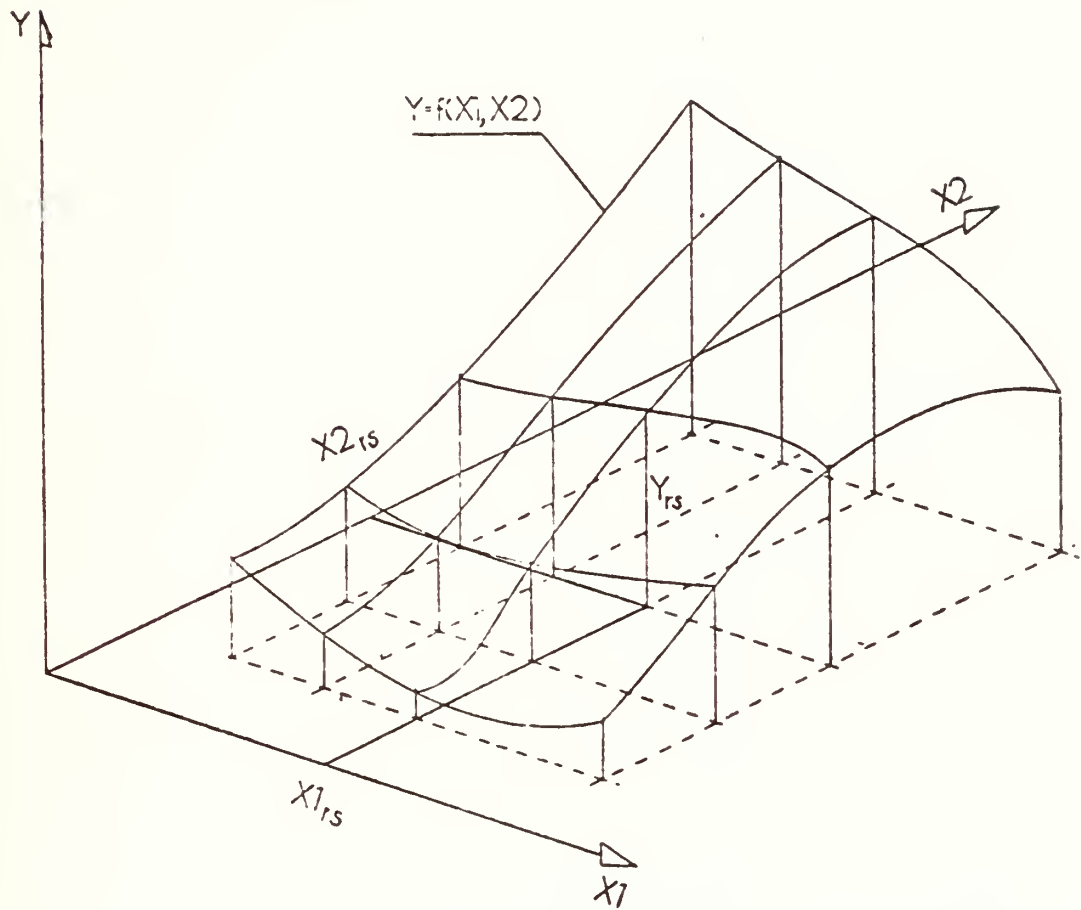
ii) graphic output



Grafic output from PROGRAM DEMO2

3. THREE DIMENSIONAL APPROXIMATION

3.1. Problem:



A data set of $NPNTS1 * NPNTS2$ data points is given, where Y depends on the parameters $X1$ and $X2$. The data pattern is to be approximated by a function $Y=f(X1, X2)$, so that the error between data points and analytically determined points is lowest.

3.2. Approach:

As for the two dimensional approximation, a polynomial, now with two independent variables X_1 and X_2 , is used.

$$Y = f(X_1, X_2)$$

$$\begin{aligned}
 Y = & C_{11} + C_{12} \cdot X_2 + C_{13} \cdot X_2^2 + \dots + C_{1M} \cdot X_2^{(M-1)} + \\
 & + [C_{21} + C_{22} \cdot X_2 + C_{23} \cdot X_2^2 + \dots + C_{2M} \cdot X_2^{(M-1)}] \cdot X_1 + \\
 & + [C_{31} + C_{32} \cdot X_2 + C_{33} \cdot X_2^2 + \dots + C_{3M} \cdot X_2^{(M-1)}] \cdot X_1^2 + \\
 & \dots + \\
 & + [C_{L1} + C_{L2} \cdot X_2 + C_{L3} \cdot X_2^2 + \dots + C_{LM} \cdot X_2^{(M-1)}] \cdot X_1^{(L-1)} \\
 Y = & \sum_{i=1}^L \left\{ \sum_{j=1}^M C_{ij} \cdot X_2^{(j-1)} \right\} \cdot X_1^{(i-1)} \quad (3.1.)
 \end{aligned}$$

The data arrangement for the three dimensional approximation already yields to the application of this method: at each constant X_1 (i.e.: $r = \text{constant}$, too) there is a data set for various X_2 . This is the case for example, when a pneumatic-velocity probe is calibrated. At each Mach number, the probe is balanced in yaw and the pitch angle is varied over the range of application. The probe output (each of two separate non-dimensional pressure differences) is a function of the Mach number and the pitch angle.

The least squares criterion becomes

$$R = \sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} [f(X1_{rs}, X2_{rs}) - Y_{rs}]^2 \right\} \quad (3.2.)$$

where the indices r and s denote the individual data points. Using equation (2.1.) R becomes

$$\begin{aligned} R = \sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} [C_{11} + C_{12} X_{rs}^1 + \dots + C_{1M} X_{rs}^{M-1} + \right. \\ + (C_{21} + C_{22} X_{rs}^1 + \dots + C_{2M} X_{rs}^{M-1}) \cdot X_{rs}^1 + \\ + (C_{31} + C_{32} X_{rs}^1 + \dots + C_{3M} X_{rs}^{M-1}) \cdot X_{rs}^2 + \\ \dots \\ \left. + (C_{L1} + C_{L2} X_{rs}^1 + \dots + C_{LM} X_{rs}^{M-1}) \cdot X_{rs}^{L-1} - Y_{rs}]^2 \right\} \end{aligned}$$

The term in the [] - bracket shall be called B.

$$R = \sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} \cdot B^2 \right\}$$

3.3. Solution:

To minimize the error, R is partially differentiated to the coefficients C_{ij} and then the partial derivatives are set to zero

$$\frac{\partial R}{\partial C_{ij}} \stackrel{!}{=} 0 \quad i = 1, \dots, L; \quad j = 1, \dots, M$$

$$\frac{\partial R}{\partial C_{ij}} = \sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} 2 \cdot B \cdot \frac{\partial B}{\partial C_{ij}} \right\} = 0$$

Assuming, that the summations extend over all data points, $\Sigma\Sigma$ should be understood to mean $\sum_{r=1}^{NPNTS1} \cdot \left\{ \sum_{s=1}^{NPNTS2} \right\}$. Performing the differentiations and rearranging the equations in matrix notation (which, while lengthy, is logically straight forward), we get, in matrix notation,

		1			2					
		(1-1)*M+1			(2-1)*M+1					
1	(1-1)*M+1	NPNTS1 * NPNTS2	$\sum \sum X_{rs}$...	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}$	$\sum \sum X_{rs}^2$...	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^2$
	...	$\sum \sum X_{rs}$	$\sum \sum X_{rs}^2$...	$\sum \sum X_{rs}$	$\sum \sum X_{rs}^2$	$\sum \sum X_{rs}^3$...	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^M$

	1*M	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^M$...	$\sum \sum X_{rs}^{2M-2}$	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^M$...	$\sum \sum X_{rs}^{2M-2}$	$\sum \sum X_{rs}^{2M-1}$
2	(2-1)*M	$\sum \sum X_{rs}$	$\sum \sum X_{rs}^2$...	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^2$	$\sum \sum X_{rs}^3$...	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^2$
	...	$\sum \sum X_{rs}^2$	$\sum \sum X_{rs}^3$...	$\sum \sum X_{rs}^M$	$\sum \sum X_{rs}^2$	$\sum \sum X_{rs}^3$...	$\sum \sum X_{rs}^M$	$\sum \sum X_{rs}^2$

	2*M	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^M$...	$\sum \sum X_{rs}^{2M-2}$	$\sum \sum X_{rs}^{M-1}$	$\sum \sum X_{rs}^M$...	$\sum \sum X_{rs}^{2M-2}$	$\sum \sum X_{rs}^{2M-1}$
	
L	(L-1)*M	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$
	...	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$...	$\sum \sum X_{rs}^{L+1}$	$\sum \sum X_{rs}^{L+2}$	$\sum \sum X_{rs}^{L+3}$...	$\sum \sum X_{rs}^{L+1}$	$\sum \sum X_{rs}^{L+2}$

	L*M	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$

A

		L								
		(L-1)*M+1			L*M					
	(L-1)*M+1	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$	C_{11}	$\sum \sum Y_{rs}$	$\sum \sum Y_{rs}$
	...	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$...	$\sum \sum X_{rs}^{L+1}$	$\sum \sum X_{rs}^{L+2}$	$\sum \sum X_{rs}^{L+3}$	C_{12}	$\sum \sum Y_{rs} \cdot X_{rs}$	$\sum \sum Y_{rs} \cdot X_{rs}$
	C_{1M}	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$
	L*M	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$	C_{21}	$\sum \sum Y_{rs} \cdot X_{rs}$	$\sum \sum Y_{rs} \cdot X_{rs}$
	(L-1)*M	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$	C_{22}	$\sum \sum Y_{rs} \cdot X_{rs} \cdot X_{rs}$	$\sum \sum Y_{rs} \cdot X_{rs} \cdot X_{rs}$
	...	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$...	$\sum \sum X_{rs}^{L+1}$	$\sum \sum X_{rs}^{L+2}$	$\sum \sum X_{rs}^{L+3}$	C_{2M}	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$
	C_{LM}	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$
	L*M	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$...	$\sum \sum X_{rs}^{L-1}$	$\sum \sum X_{rs}^L$	$\sum \sum X_{rs}^{L+1}$	C_{LM}	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$	$\sum \sum Y_{rs} \cdot X_{rs}^{M-1}$

* C = B (3.3.)

3.4. Equation System:

3.4.1. System Matrix A

This system matrix needs some examination. A crucial step is the introduction of submatrices 1. class into the system matrix.

$$A = \begin{vmatrix} A_{11}^* & A_{12}^* & A_{1L}^* \\ A_{21}^* & A_{22}^* & A_{2L}^* \\ A_{L1}^* & A_{L2}^* & A_{LL}^* \end{vmatrix}$$

The asterisk * denotes a submatrix 1. class where

$$A_{11}^* = \begin{vmatrix} NPNTS1 * NPNTS2 & \sum \sum x_{15} & \sum \sum x_{15}^{M-1} \\ \sum \sum x_{15} & \sum \sum x_{15}^2 & \sum \sum x_{15}^M \\ \sum \sum x_{15}^{M-1} & \sum \sum x_{15}^M & \sum \sum x_{15}^{2M-2} \end{vmatrix}$$

$$A_{12}^* = \begin{vmatrix} \sum \sum x_{15} & \sum \sum x_{15} x_{25} & \sum \sum x_{15} x_{25}^{M-1} \\ \sum \sum x_{15} x_{25} & \sum \sum x_{15} x_{25}^2 & \sum \sum x_{15} x_{25}^M \\ \sum \sum x_{15} x_{25}^{M-1} & \sum \sum x_{15} x_{25}^M & \sum \sum x_{15} x_{25}^{2M-2} \end{vmatrix}$$

$$A_{13}^* = \begin{vmatrix} \sum \sum x_{15}^2 & \sum \sum x_{15}^2 x_{25} & \sum \sum x_{15}^2 x_{25}^{M-1} \\ \sum \sum x_{15}^2 x_{25} & \sum \sum x_{15}^2 x_{25}^2 & \sum \sum x_{15}^2 x_{25}^M \\ \sum \sum x_{15}^2 x_{25}^{M-1} & \sum \sum x_{15}^2 x_{25}^M & \sum \sum x_{15}^2 x_{25}^{2M-2} \end{vmatrix}$$

etc.

Now it can be seen that all submatrices on the same diagonal bands

2. order are identical. The submatrices A_{ij}^* subsequently are renamed A_k^*

$$A_1^* = A_{11}^*$$

$$A_2^* = A_{21}^* = A_{12}^*$$

$$A_3^* = A_{31}^* = A_{22}^* = A_{13}^*$$

$$A_4^* = A_{41}^* = A_{32}^* = A_{23}^* = A_{14}^*$$

⋮

$$A_L^* = A_{L1}^* = A_{L-1,2}^* = A_{L-2,3}^* = \dots = A_{2,L-1}^* = A_{1,L}^*$$

$$A_{L+1}^* = A_{L,2}^* = A_{L-1,3}^* = \dots = A_{3,L-1}^* = A_{2,L}^*$$

$$A_{L+2}^* = A_{L,3}^* = \dots = A_{4,L-1}^* = A_{3,L}^*$$

⋮

$$A_{2L-1}^* = A_{L,L}^*$$

where A_k^* can be written as

$$A_k^* = \begin{vmatrix} \sum \sum x_{rs}^{k-1} x_{rs}^0 & \sum \sum x_{rs}^{k-1} x_{rs}^1 & \sum \sum x_{rs}^{k-1} x_{rs}^{M-1} \\ \sum \sum x_{rs}^{k-1} x_{rs}^1 & \sum \sum x_{rs}^{k-1} x_{rs}^2 & \sum \sum x_{rs}^{k-1} x_{rs}^M \\ \sum \sum x_{rs}^{k-1} x_{rs}^{M-1} & \sum \sum x_{rs}^{k-1} x_{rs}^M & \sum \sum x_{rs}^{k-1} x_{rs}^{2M-2} \end{vmatrix}$$

$k = 1, \dots, 2L-1$

Like the system matrix for the two dimensional approximation this submatrix 1. class is not only symmetrical to its main diagonal line 1. order, but also the elements on each diagonal line 2. order are identical. The elements $a_{k;ij}$ of the submatrix A_k therefore are

renamed to $a_{k;L}$, where k specifies the place of the submatrix in the system matrix and i and j define the place of $a_{k;ij}$ in the submatrix A_k^* .

$$\begin{aligned}
 a_{k;1} &= a_{k;11} \\
 a_{k;2} &= a_{k;21} = a_{k;12} \\
 a_{k;3} &= a_{k;31} = a_{k;22} = a_{k;13} \\
 &\vdots \\
 a_{k;M} &= a_{k;M1} = a_{k;M-1,2} = a_{k;M-2,3} = \dots = a_{k;2,M-1} = a_{k;1,M} \\
 a_{k;M+1} &= a_{k;M2} = a_{k;M-1,3} = \dots = a_{k;3,M-1} = a_{k;2,M} \\
 a_{k;M+2} &= a_{k;M3} = \dots = a_{k;4,M-1} = a_{k;3,M} \\
 &\vdots \\
 a_{k;2M-1} &= a_{k;MM}
 \end{aligned}$$

where $a_{k;L}$ can be written as

$$a_{k;L} = \left\{ \sum_{r=1}^{NPNTS1} \left\{ \sum_{s=1}^{NPNTS2} X1_{rs}^{k-1} X2_{rs}^{L-1} \right\} \right\}$$

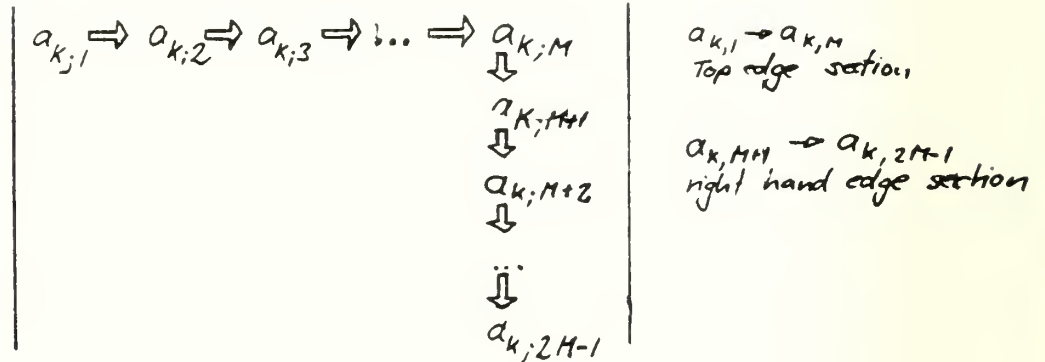
$$k = 1, \dots, 2L-1 ; \quad L = 1, \dots, 2M-1$$

(3.4.)

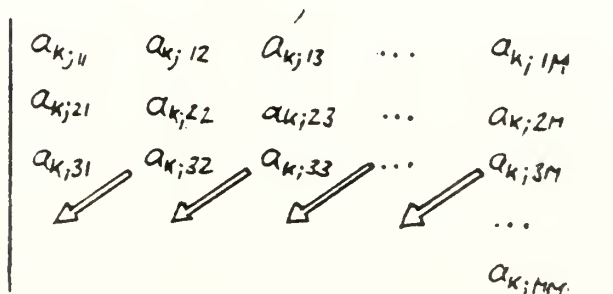
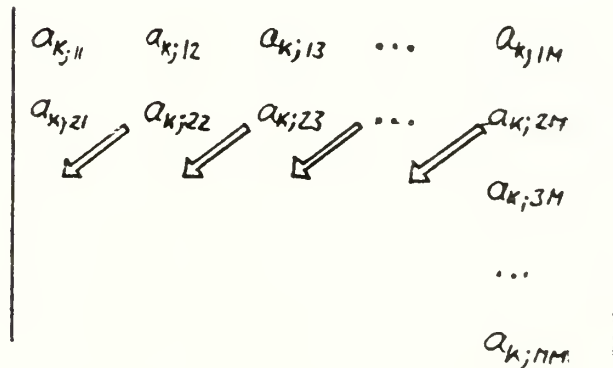
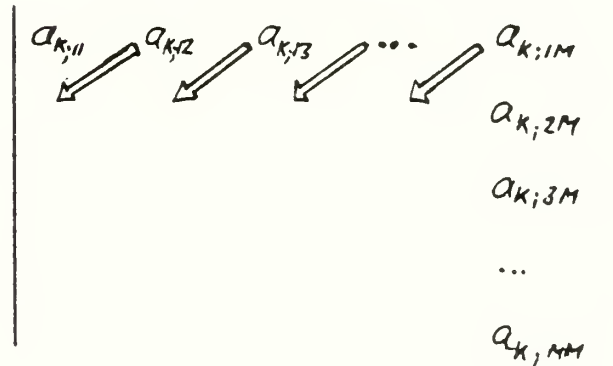
Equation (3.4.) therefore is the fundamental step towards the programming of the system matrix and is programmed in REAL FUNCTION S3. Since the data $X1$, $X2$ and Y are already provided through a COMMON block named DATA3, only NPNTS1, NPNTS2, IPOWR1 (= $k-1$) and IPOWR2 (= $L-1$) have to be passed to the function that returns the value of $a_{k;L}$ through the function name S3.

SUBROUTINE MAT31 presets a submatrix A_k^* in the following way:

i) Preset edge section elements (using REAL FUNCTION S3)



ii) Copy defined elements diagonally



$$A_K^* = \begin{vmatrix} a_{k,11} & a_{k,12} & a_{k,13} & \dots & a_{k,1M} \\ a_{k,21} & a_{k,22} & a_{k,23} & \dots & a_{k,2M} \\ a_{k,31} & a_{k,32} & a_{k,33} & \dots & a_{k,3M} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & a_{k,MM} \end{vmatrix}$$

$$A_K^* = \begin{vmatrix} a_{k,11} & a_{k,12} & a_{k,13} & \dots & a_{k,1M} \\ a_{k,21} & a_{k,22} & a_{k,23} & \dots & a_{k,2M} \\ a_{k,31} & a_{k,32} & a_{k,33} & \dots & a_{k,3M} \\ \dots & \dots & \dots & \dots & \dots \\ a_{k,M1} & a_{k,M2} & a_{k,M3} & \dots & a_{k,MM} \end{vmatrix}$$

This subroutine that needs NPNTS1, NPNTS2, M and IPOWR1 (= k-1) as input parameters, returns A_K^* (SUBM1(7,7)) to SUBROUTINE MAT3, which presets the entire system matrix in the following way

- i) Preset edge section submatrices (using SUBROUTINE MAT31)

$$\left| \begin{array}{c} A_1^* \Rightarrow A_2^* \Rightarrow \dots \Rightarrow A_L^* \\ \downarrow \\ A_{L+1}^* \\ \downarrow \\ \vdots \\ \downarrow \\ A_{2L-1}^* \end{array} \right|$$

$A_1^* \rightarrow A_L^*$
 Top edge section
 $A_{L+1}^* \rightarrow A_{2L-1}^*$
 right hand edge section

ii) Copy defined submatrices diagonally

$$\begin{vmatrix} A_{11}^* & A_{12}^* & \dots & A_{1L}^* \\ & A_{22}^* & & A_{2L}^* \\ & & & \dots \\ & & & A_{LL}^* \end{vmatrix}$$

$$\begin{vmatrix} A_{11}^* & A_{12}^* & \dots & A_{1L}^* \\ A_{21}^* & A_{22}^* & \dots & A_{2L}^* \\ & & & \dots \\ & & & A_{LL}^* \end{vmatrix}$$

$$\begin{vmatrix} A_{11}^* & A_{12}^* & \dots & A_{1L}^* \\ A_{L1}^* & A_{L2}^* & \dots & A_{LL}^* \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_{LL}^* \end{vmatrix}$$

$$A = \begin{vmatrix} A_{11}^* & A_{12}^* & \dots & A_{1L}^* \\ A_{21}^* & A_{22}^* & \dots & A_{2L}^* \\ \dots & \dots & \dots & \dots \\ A_{L1}^* & A_{L2}^* & \dots & A_{LL}^* \end{vmatrix}$$

The parameters to call this subroutine are NPNTS1 and NPNTS2 (# of data points), L and M (<the order of the polynomial fit + 1> in X1 and X2 - direction) and IPRINT. The coefficients C_{ij} are returned.

3.4.2. Right hand side vector B

The right hand side vector B will be divided up into L subvectors 1. class.

$$B = \begin{vmatrix} B_1^* \\ B_2^* \\ \dots \\ B_L^* \end{vmatrix}$$

Again, the asterisk (*) denotes a subvector 1. class. Now the subvectors 1. class B_k^* can be written as

$$B_k^* = \begin{vmatrix} \sum \sum Y_{rs} \cdot X1_{rs}^{k-1} \\ \sum \sum Y_{rs} \cdot X1_{rs}^{k-1} \cdot X2_{rs} \\ \dots \\ \sum \sum Y_{rs} \cdot X1_{rs}^{k-1} \cdot X2_{rs}^M \end{vmatrix}$$

$$k = 1, \dots, L$$

Finally, the elements of the subvectors 1. class B_k^* can be written as

$$b_{k;l} = \sum \sum Y_{rs} \cdot X1_{rs}^{k-1} \cdot X2_{rs}^{l-1}$$

$$k = 1, \dots, L; \quad l = 1, \dots, M$$

To compute $b_{k;l}$, REAL FUNCTION S3 is used. Data X1, X2 and Y are available through a COMMON block named DATA3. NPNTS1, NPNTS2, IPOWR1 (= k-1), IPOWR2 (= L-1) and IY (=1) have to be passed to the function and the value of $b_{k;l}$ is returned through the function name S3.

Using FORTRAN programming language, the allocation of $b_{k;L}$ can be performed by two stacked DO loops.

```
      I=0
      DO 08 L1=1,L,1
        IPOWR1=L1-1
        DO 08 M1=1,M,1
          IPOWR2=M1-1
          I=I+1
08    B(I) = S3(NPNTS1,NPNTS2,IPOWR1,IPOWR2,1)
```

3.5. Software:

The software to compute the coefficients for a three dimensional approximation is described in Appendix A and is implemented in the TPL HP-21MX computer system. To work correctly with these program modules, the user has to conform to the following conventions:

- i) Provide the data in three arrays (Type: REAL) of 16 * 16 elements through a COMMON block named DATA3.

```
COMMON / DATA3 / X1, X2, Y
REAL X1 (16,16), X2(16,16), Y(16,16)
```

- ii) Dimension an array (Type: REAL) of 7*7 elements to contain the coefficients.

```
REAL COEF (7,7)
```

- iii) Define the parameters NPNTS1, NPNTS2, L, M and IPRINT.

```
NPNTS1 ... # of X1 variations
          L ≤ NPNTS1 ≤ 16
```

```
NPNTS2 ... # of X2 variations
          M ≤ NPNTS2 ≤ 16
```

```
L ... (desired order of approximation
        polynomial w.r.t. X1) + 1
```

```
M ... (desired order of approximation
        polynomial w.r.t. X2) + 1
```

```
IPRINT ... controls quantity of print out
          2 ... Print system matrix and
                right hand side vector
                before and after Gauss
                Jordan Elimination
          1 ... Print equation system after
                Gauss Jordan Elimination
          <0 ... No print out
          >0 ... Display equation (3.1.)
                with the actual parameters
```

- iv) If a user program uses subroutine MAT3, the software modules have to be loaded using the procedure outlined in section 2.4. iv.

If all these requirements are met, the correct call for the subroutine is:

```
CALL MAT (NPNTS1,NPNTS2,L,M,COEF,IPRINT)
```

Upon completed execution of this approximation routine the array COEF contains the coefficients. Externals used by MAT3 are: AB3, DATA3, MAT31, IEL3, S3. Under no circumstances may the user use any of these names for modules of his own program.

3.6. Sample User Program

3.6.1. FTN4 Compiler listing of sample user program

```

0001 FTN4,L
0002 PROGRAM LAB (3,99)
0003 * ,Plot calibration points and calculate coefficients.
0004
0005 COMMON / AFLD / PLOTR
0006 COMMON / DATA3 / X1,X2,Y
0007 COMMON / AB3 / A,B
0008
0009 REAL PLOTR(256)
0010 REAL X1(16,16),X2(16,16),Y(16,16)
0011 REAL A(49,49),B(49)
0012
0013 REAL YT(16,16),R(16),COEFF(7,7)
0014 INTEGER IDCB(144),IFILE(3),NOCR(2),ICLR(3),YTITLE(4),ZTITLE(4)
0015
0016 DATA PI /3.141593/
0017 DATA NOLF /006537B/
0018 DATA NOCR /000033B,040433B/
0019 DATA ICLR /015524B,015515B,006537B/
0020
0021 C FORMATS LAB START
0022 121 FORMAT (" Enter raw data file name "2A2)
0023 122 FORMAT (3A2)
0024 123 FORMAT (" Select 1 ... Pitch angle or 2 .
0025 * .. Velocity X "2A2)
0026 124 FORMAT (" Select 1 ... Pitch angle v/s Gamma and Beta (
0027 * = ( , ) )"/
0028 * 2 ... "1H" " "1H"," v/s Gamma a
0029 *nd Delta ( = ( ) ) "2A2)
0030 125 FORMAT (" Select 1 ... Velocity v/s Gamma and Beta ( X=
0031 *X( , ) )"/
0032 * "2 ... "1H"," v/s Gamma and Delta
0033 * ( X=X( , ) ) "2A2)
0034 126 FORMAT (" Enter # of Mach Numbers and # of pitch angle
0035 *s "2A2)
0036 160 FORMAT (" Enter 1 to print input data "2A2)
0037 131 FORMAT (/ " Calibrate = ( , )"/)
0038 132 FORMAT (/ " Calibrate = ( , )"/)
0039 133 FORMAT (/ " Calibrate X=X( , )"/)
0040 134 FORMAT (/ " Calibrate X=X( , )"/)
0041 127 FORMAT (24X"9X"9X" ")
0042 128 FORMAT (24X"9X"9X" ")
0043 129 FORMAT (24X"9X"9X"X")
0044 130 FORMAT (24X"9X"9X"X")
0045 150 FORMAT (" minimum values",3(1X,F9.3))
0046 151 FORMAT (" maximum values",3(1X,F9.3))
0047 152 FORMAT (" of user coordinates"3X"X1"8X"X2"9X"Y")
0048 153 FORMAT (" Enter 1 to redefine these data "2A2)
0049 154 FORMAT (" Enter min, max, min, "
0050 * " max, # partitions, A ")
0051 155 FORMAT (" Enter min, max, min, "
0052 * " max, # partitions, A ")
0053 156 FORMAT (" Enter min, max, min, max, X min, X ma
0054 *x, # partitions, A X")
0055 157 FORMAT (" Enter min, max, min, max, X min, X ma
0056 *x, # partitions, A X")
0057 158 FORMAT (1X,3(F8.3,1X,F8.3,5X),3I4)
0058 101 FORMAT (" Enter order M of X1 - approximation "2A2)
0059 102 FORMAT (" Enter order N of X2 - approximation "2A2)
0060 107 FORMAT (" Enter IPRINT "2A2)
0061 105 FORMAT (" Select 1 ... to calculate the absolute error"/8X
0062 * "2 "4H" "9X"the relative error "2A2)
0063 103 FORMAT (" Change coefficients ... 1"/"2A2)
0064 * " Change axes "2 "2A2)
0065 104 FORMAT (" Enter ALPHA, ALPHA, X0 and Y0")
0066 149 FORMAT ("((3A2)))
0067 611 FORMAT ("1"//// " These data from file "3A
0068 612 FORMAT (/16X,I2", Mach number"/)
0069 613 FORMAT (6X"Gamma"6X"Beta"7X"Phi"/)
0070 614 FORMAT (6X"Gamma"5X"Delta"7X"Phi"/)
0071 615 FORMAT (6X"Gamma"6X"Beta"5X"X vel"/)
0072 616 FORMAT (6X"Gamma"5X"Delta"5X"X vel"/)
0073 617 FORMAT (1X,4(1X,F9.3))
0074 618 FORMAT (/9X"X1"8X"X2"9X"Y")
0075 602 FORMAT ((3X,16(6X,I2)))
0076 603 FORMAT (1X,I2,16(1X,F7.3)/3(3X,16(1X,F7.3)/))

```

```

0077      604 FORMAT (///" Absolute error at each point ( error = (calculated
0078      *value - measured value) )"/)
0079      644 FORMAT (///" Relative error at each point in % ( error = (calculated
0080      *value - measured value)/measured value )"/)
0081      606 FORMAT (1X,I2,18(1X,F6.1)/3(3X,18(1X,F7.3)/))
0082      1101 FORMAT (8H Beta )
0083      1102 FORMAT (8H Delta )
0084      1103 FORMAT (8H Phi )
0085      1104 FORMAT (8H X vel )
0086      LI = LOGLU(I)
0087      LO = 6
0088
0089
0090
0091      C .....
0092      C :
0093      C : read raw data file.
0094      C :
0095      C .....
0096      41 WRITE (LI, 121) NOCR
0097      READ (LI, 122) IFILE
0098      WRITE (LI, 149) ICLR
0099      42 WRITE (LI, 123) NOCR
0100      READ (LI, *) IPX
0101      WRITE (LI, 149) ICLR
0102      IF (IPX.LT.1 .OR. IPX.GT.2) GO TO 42
0103      43 CONTINUE
0104      IF (IPX.EQ.1) WRITE (LI, 124) NOCR
0105      IF (IPX.EQ.2) WRITE (LI, 125) NOCR
0106      READ (LI, *) IBD
0107      WRITE (LI, 149) (ICLR,I=1,2,1)
0108      IF (IBD.LT.1 .OR. IBD.GT.2) GO TO 43
0109      IF (IPX.EQ.1) IRECY=13
0110      IF (IPX.EQ.2) IRECY=9
0111      IF (IPX.EQ.1 .AND. IBD.EQ.1 ) WRITE (LI, 131)
0112      IF (IPX.EQ.1 .AND. IBD.EQ.2 ) WRITE (LI, 132)
0113      IF (IPX.EQ.2 .AND. IBD.EQ.1 ) WRITE (LI, 133)
0114      IF (IPX.EQ.2 .AND. IBD.EQ.2 ) WRITE (LI, 134)
0115      CALL OPEN (IDCB,IERR,IFILE,IOPTN,0,28,144)
0116      IF (IERR.LT.0) GO TO 41
0117      CALL READF (IDCB,IERR,X1,512,LEN,1)
0118      IF ( IERR .LT. 0 ) STOP 0002
0119      CALL READF (IDCB,IERR,X2,512,LEN,5)
0120      IF ( IERR .LT. 0 ) STOP 0003
0121      CALL READF (IDCB,IERR,YT,512,LEN,IRECY)
0122      IF ( IERR .LT. 0 ) STOP 0004
0123      CALL CLOSE (IDCB,IERR,0)
0124      44 WRITE (LI, 126) NOCR
0125      READ (LI, *) NMACH,NPITCH
0126      WRITE (LI, 149) ICLR
0127      IF (NMACH.GT.16) GO TO 44
0128      IF (NPITCH.GT.16) GO TO 44
0129      DO 801 I=1,NMACH,1
0130      DO 801 J=1,NPITCH,1
0131      801 Y(I,J)=YT(I,J)
0132
0133
0134      C .....
0135      C :
0136      C : Correct input data and optional output of data, coefficient
0137      C : are based upon.
0138      C :
0139      C .....
0140      C .....
0141      IF (IBD.EQ.2) GO TO 48
0142      DO 47 I=1,NMACH,1
0143      DO 47 J=1,NPITCH,1
0144      47 X2(I,J)=X2(I,J)/X1(I,J)
0145      48 WRITE (LI, 160) NOCR
0146      READ (LI, *) IDUM
0147      WRITE (LI, 149) ICLR
0148      IF (IDUM.NE.1) GO TO 55
0149      WRITE (LO, 611) IFILE
0150      DO 54 I=1,NMACH,1
0151      WRITE (LO, 612) I
0152      IF (IPX.EQ.1 .AND. IBD.EQ.1) WRITE (LO, 613)

```

```

0153 IF (IPX.EQ.1 .AND. IRD.EQ.2) WRITE (LO, 614)
0154 IF (IPX.EQ.2 .AND. IRD.EQ.1) WRITE (LO, 615)
0155 IF (IPX.EQ.2 .AND. IRD.EQ.2) WRITE (LO, 616)
0156 DO 54 J=1, NPITCH, 1
0157 54 WRITE (LO, 617) X1(I,J), X2(I,J), Y(I,J)
0158 WRITE (LO, 618)
0159
0160
0161
0162 C .....
0163 C . Calculate minimum and maximum values for all coordinates and .
0164 C . optional redefinition of minimum and maximum values by user. .
0165 C .....
0166 C
0167 C
0168 55 XUMAX=X1(1,1)
0169 XUMIN=X1(1,1)
0170 YUMAX=X2(1,1)
0171 YUMIN=X2(1,1)
0172 ZUMAX=Y(1,1)
0173 ZUMIN=Y(1,1)
0174 DO 45 I=1, NMACH, 1
0175 DO 45 J=1, NPITCH, 1
0176 IF (X1(I,J).GT.XUMAX) XUMAX=X1(I,J)
0177 IF (X1(I,J).LT.XUMIN) XUMIN=X1(I,J)
0178 IF (X2(I,J).GT.YUMAX) YUMAX=X2(I,J)
0179 IF (X2(I,J).LT.YUMIN) YUMIN=X2(I,J)
0180 IF (Y(I,J).GT.ZUMAX) ZUMAX=Y(I,J)
0181 IF (Y(I,J).LT.ZUMIN) ZUMIN=Y(I,J)
0182 45 CONTINUE
0183 IF (IPX.EQ.1 .AND. IRD.EQ.1) WRITE (LI, 127)
0184 IF (IPX.EQ.1 .AND. IRD.EQ.2) WRITE (LI, 128)
0185 IF (IPX.EQ.2 .AND. IRD.EQ.1) WRITE (LI, 129)
0186 IF (IPX.EQ.2 .AND. IRD.EQ.2) WRITE (LI, 130)
0187 WRITE (LI, 150) XUMIN, YUMIN, ZUMIN
0188 WRITE (LI, 151) XUMAX, YUMAX, ZUMAX
0189 WRITE (LI, 152)
0190 INCX = 4
0191 INCY = 4
0192 INCZ = 4
0193 WRITE (LI, 153) NOCR
0194 READ (LI, *) I
0195 WRITE (LI, 149) ICLR
0196 IF (I.NE.1) GO TO 49
0197 IF (IPX.EQ.1 .AND. IRD.EQ.1) WRITE (LI, 154)
0198 IF (IPX.EQ.1 .AND. IRD.EQ.2) WRITE (LI, 155)
0199 IF (IPX.EQ.2 .AND. IRD.EQ.1) WRITE (LI, 156)
0200 IF (IPX.EQ.2 .AND. IRD.EQ.2) WRITE (LI, 157)
0201 READ (LI, *) XUMIN, XUMAX, YUMIN, YUMAX, ZUMIN, ZUMAX, INCX, INCY, INCZ
0202 WRITE (LI, 158) XUMIN, XUMAX, YUMIN, YUMAX, ZUMIN, ZUMAX, INCX, INCY, INCZ
0203
0204
0205 C .....
0206 C . Initialize plotter; define plotter area; calculate scaling .
0207 C . coefficients XA, XB, YA, YB, ZA and ZB. .
0208 C .....
0209 C
0210 C
0211 C
0212 49 CALL INITG (13)
0213 XPMIN = 0.
0214 XPMAX = 10.
0215 YPMIN = 0.
0216 YPMAX = 10.
0217 ZPMIN = 0.
0218 ZPMAX = 10.
0219 ALPHAX = 330.
0220 ALPHAY = 30.
0221 ALPHAZ = 90.
0222 XA = (XPMAX - XPMIN) / (XUMAX - XUMIN)
0223 XB = (XPMIN * XUMAX - XPMAX * XUMIN) / (XUMAX - XUMIN)
0224 XL = (XPMAX - XPMIN)
0225 YA = (YPMAX - YPMIN) / (YUMAX - YUMIN)
0226 YB = (YPMIN * YUMAX - YPMAX * YUMIN) / (YUMAX - YUMIN)
0227 YL = (YPMAX - YPMIN)
0228 ZA = (ZPMAX - ZPMIN) / (ZUMAX - ZUMIN)

```



```

0229      ZB      = (ZPMIN*ZUMAX-ZPMAX*ZUMIN)/(ZUMAX-ZUMIN)
0230      ZL      = (ZPMAX      -ZPMIN      )
0231      XO      = 8.
0232      YO      = 12.
0233      85      PLOTR(61) = (ALPHAX*PI)/180.
0234              PLOTR(62) = (ALPHAY*PI)/180.
0235              PLOTR(63) = (ALPHAZ*PI)/180.
0236              PLOTR(64) = XO
0237              PLOTR(65) = YO
0238              PLOTR(66) = XA
0239              PLOTR(67) = XB
0240              PLOTR(68) = YA
0241              PLOTR(69) = YB
0242              PLOTR(70) = ZA
0243              PLOTR(71) = ZB
0244              GO TO (86,87) IBD
0245      86      CALL CODE
0246              WRITE (YTITLE,1101)
0247              GO TO 88
0248      87      CALL CODE
0249              WRITE (YTITLE,1102)
0250      88      GO TO (301,302) IPX
0251      301     CALL CODE
0252              WRITE (ZTITLE,1103)
0253              GO TO 303
0254      302     CALL CODE
0255              WRITE (ZTITLE,1104)
0256      303     CALL SETSM (113,1.)
0257              CALL AXIS (XO,YO,XL,ALPHAX,6H Gamma, 8,XUMIN,XUMAX,4HF6.2,6,INCX)
0258              CALL AXIS (XO,YO,YL,ALPHAY,YTITLE,-8,YUMIN,YUMAX,4HF6.2,6,INCY)
0259              CALL AXIS (XO,YO,ZL,ALPHAZ,ZTITLE,-8,ZUMIN,ZUMAX,4HF6.2,6,INCZ)
0260
0261
0262
0263
0264      C .....
0265      C .   Plot measured calibration surface.
0266      C .
0267      C .....
0268      CALL SETSM (113,2.)
0269      DO 01 I=1,NMACH,1
0270      DO 01 J=1,NPITCH,1
0271      CALL THRTW (XPLOT,YPLOT,X1(I,J),X2(I,J),Y(I,J))
0272      IF ( J .EQ. 1 ) CALL PLOT (XPLOT,YPLOT,2)
0273      IF ( J .GT. 1 ) CALL PLOT (XPLOT,YPLOT,3)
0274      01      CONTINUE
0275      DO 02 J=1,NPITCH,1
0276      DO 02 I=1,NMACH,1
0277      CALL THRTW (XPLOT,YPLOT,X1(I,J),X2(I,J),Y(I,J))
0278      IF ( I .EQ. 1 ) CALL PLOT (XPLOT,YPLOT,2)
0279      IF ( I .GT. 1 ) CALL PLOT (XPLOT,YPLOT,3)
0280      02      CONTINUE
0281
0282
0283
0284      C .....
0285      C .   Calculate calibration surface coefficients.
0286      C .
0287      C .....
0288
0289      91      CALL SETSM (113,0.)
0290              WRITE (LI, 101) NOCR
0291              READ (LI, *) MORDER
0292              WRITE (LI, 149) ICLR
0293              IF (MORDER.GT.6) GO TO 91
0294      92      WRITE (LI, 102) NOCR
0295              READ (LI, *) NORDER
0296              WRITE (LI, 149) ICLR
0297              WRITE (LI, 107) NOCR
0298              READ (LI, *) IPRINT
0299              WRITE (LI, 149) ICLR
0300              IF (NORDER.GT.6) GO TO 92
0301              M=MORDER+1
0302              N=NORDER+1
0303              CALL MAT3 (NMACH,NPITCH,M,N,COEFF,IPRINT)
0304

```



```

0305
0306
0307 C .....
0308 C .....
0309 C : Overwrite data array with calculated data, based on the
0310 C : just obtained coefficients.
0311 C .....
0312 C .....
0313 411 WRITE (LI, 105) NOCR
0314 READ (LI, *) IAR
0315 WRITE (LI, 149) (ICLR, I=1, 2)
0316 IF (IAR.LT.1 .OR. IAR.GT.2) GO TO 411
0317 IF (IAR.EQ.1) WRITE (LO, 604)
0318 IF (IAR.EQ.2) WRITE (LO, 644)
0319 WRITE (LO, 602) (J, J=1, NPITCH, 1)
0320 DO 20 I=1, NMACH, 1
0321 DO 19 J=1, NPITCH, 1
0322 SUM=0.
0323 DO 96 I1=1, M, 1
0324 IEXP1=I1-1
0325 IF (IEXP1.EQ.0) GO TO 401
0326 X1EXP=X1(I, J)**IEXP1
0327 GO TO 402
0328 401 X1EXP=1.
0329 402 CONTINUE
0330 DO 96 J1=1, N, 1
0331 IEXP2=J1-1
0332 IF (IEXP2.EQ.0) GO TO 403
0333 X2EXP=X2(I, J)**IEXP2
0334 GO TO 404
0335 403 X2EXP=1.
0336 404 CONTINUE
0337 96 SUM=SUM+COEFF(I1, J1)*X1EXP*X2EXP
0338 GO TO (412, 413) IAR
0339 412 R(J)=SUM-Y(I, J)
0340 GO TO 19
0341 413 R(J)=(SUM-Y(I, J))/Y(I, J)*100.
0342 19 Y(I, J)=SUM
0343 20 WRITE (6, 603) I, (R(J), J=1, NPITCH, 1)
0344 WRITE (6, 602) (J, J=1, NPITCH, 1)
0345
0346
0347 C .....
0348 C .....
0349 C : Plot calculated calibration surface.
0350 C .....
0351 C .....
0352 C .....
0353 CALL SETSM (113, 3.)
0354 DO 81 I=1, NMACH, 1
0355 DO 81 J=1, NPITCH, 1
0356 CALL THRTW (XPLT, YPLT, X1(I, J), X2(I, J), Y(I, J))
0357 IF (J.EQ.1) CALL PLOT (XPLT, YPLT, 2)
0358 IF (J.GT.1) CALL PLOT (XPLT, YPLT, 3)
0359 81 CONTINUE
0360 DO 82 J=1, NPITCH, 1
0361 DO 82 I=1, NMACH, 1
0362 CALL THRTW (XPLT, YPLT, X1(I, J), X2(I, J), Y(I, J))
0363 IF (I.EQ.1) CALL PLOT (XPLT, YPLT, 2)
0364 IF (I.GT.1) CALL PLOT (XPLT, YPLT, 3)
0365 82 CONTINUE
0366
0367
0368 C .....
0369 C .....
0370 C : Redefine Y.
0371 C .....
0372 C .....
0373 C .....
0374 DO 802 I=1, NMACH, 1
0375 DO 802 J=1, NPITCH
0376 802 Y(I, J)=YT(I, J)
0377
0378
0379 C .....
0380 C .....

```

```

0381 C      :
0382 C      : Next step ?
0383 C      :
0384 C      : .....
0385      WRITE (LI,103) NOCR
0386      READ  (LI, *) IDUM
0387      WRITE (LI, 149) (ICLR,I=1,2,1)
0388      IF (IDUM.EQ.1) GO TO 91
0389      IF (IDUM.EQ.2) GO TO 89
0390      CALL STOPG
0391      STOP 077
0392      89 WRITE (LI,104)
0393      READ  (LI, *) ALPHAX,ALPHAY,XO,YO
0394      WRITE (LI, 149) ICLR
0395      GO TO 85
0396      END

```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 04465 COMMON = 00000

3.6.2. Load map

LAB 10042 20622 Plot calibration points and calculate coefficients.

to load these program modules, enter (from LOADR): MS,%TPLBL

DATA3	20623	23622	3D-Approximation / DATA3 /
MAT3	23623	27455	3D-Approximation arrange system matrix and vector.
MAT31	27456	27651	3D-Approximation arrange submatrix 1. class.
S3	27652	30070	3D-Approximation compute summations.
AB3	30071	41534	3D-Approximation / AB3 /

AFLD	41535	42534	GSP / AFLD / A(256)	
AXIS	42535	45054	GSP	Draw and label axes.
INITG	45055	46067	GSP	read control array A; initialize plotter.
PLOT	46070	46620	GSP	move pen to a defined point.
SETSM	46621	47561	GSP	change plotter modes.
STOPG	47562	47624	GSP	terminate graphics.
THRTW	47625	50111	GSP	converts 3D user coordinates to plotter units.
FCTR	50112	50115	GSP / FCTR / FX,FY	
FCTR	50116	50135	GSP	vary size of the plot.

LOGLU	50136	50213	92067-16268	REV.1903	790228
READF	50214	51155	92067-16125	REV.1940	790719
OPEN	51156	51453	92067-16125	REV.1903	790215
CLOSE	51454	51663	92067-16125	REV.1903	781229
CLRIO	51664	51672	750701	24998-16001	
OVRD	51673	51673	92067-16125	REV.1903	780526
\$SMVE	51674	51762	92067-16268	REV.1903	790202
LURQ	51763	52345	92067-16268	REV.1903	790223
.DADS	52346	52455	780818	24998-16001	
.DMP	52456	52623	780818	24998-16001	
.DDI	52624	53124	781021	24998-16001	
SESSN	53125	53142	92067-16125	REV.1903	780413
R/W\$	53143	53301	92067-16125	REV.1903	781214
P.PAS	53302	53330	92067-16125	REV.1903	740801
.DNG	53331	53340	780818	24998-16001	
PAUSE	53341	53441	771122	24998-16001	
\$ALRN	53442	53557	92067-16268	REV.1903	770715
FMTIO	53560	55056	24998-16002	REV.1926	790417
ERR0	55057	55146	771122	24998-16001	
TAN	55147	55253	780424	24998-16001	
ABS	55254	55262	750701	24998-16001	
.SNCS	55263	55424	780424	24998-16001	
.DDE	55425	55436	780818	24998-16001	
.DIN	55437	55444	780818	24998-16001	
.RTOI	55445	55540	780921	24998-16001	
.FPWR	55541	55602	781106	24998-16001	
.SBT	55603	55643	770518	24998-16001	
.FCM	55644	55660	750701	24998-16001	
PAU.E	55661	55661	750701	24998-16001	
ERO.E	55662	55662	750701	24998-16001	
.CMRS	55663	55746	780424	24998-16001	
\$OPEN	55747	56123	92067-16125	REV.1903	790103
RW\$UB	56124	56471	92067-16125	REV.1903	781003
RWND\$	56472	56614	92067-16125	REV.1903	780801
FRMTR	56615	62252	24998-16002	REV.1926	790503
FMT.E	62253	62253	24998-16002	REV.1901	781107
IB4A2	62254	63554	59310-1X013	REV.1940	790802 1153
REIO	63555	63701	92067-16268	REV.1903	790316
RMPAR	63702	63744	781106	24998-16001	
PNAME	63745	64012	771121	24998-16001	
LUTRU	64013	64121	92067-16268	REV.1903	790223
IPUT	64122	64142	92067-16125	REV.1903	740801
\$SETP	64143	64167	781106	24998-16001	
.CFER	64170	64245	750701	24998-16001	
.LBT	64246	64276	770518	24998-16001	

24 PAGES RELOCATED 24 PAGES REQ'D NO PAGES EMA NO PAGES MSEG
 LINKS:KP PROGRAM:LR LOAD:TE COMMON:NC
 /LOADR:LAB READY AT 3:10 PM FRI., 26 SEPT, 1980
 /LOADR:\$END

3.6.3. Results

i) printed output

These data from file CALLAA.

1. Mach number

Gamma	Beta	Phi
-.810	-.005	-35.000
-.519	-.005	-25.000
-.280	-.005	-15.000
-.182	-.005	-10.000
-.103	-.006	-5.000
-.059	-.006	-2.500
-.018	-.006	0.000
.019	-.006	2.500
.054	-.006	5.000
.133	-.006	10.000
.197	-.006	15.000
.360	-.006	25.000
.605	-.006	35.000

2. Mach number

Gamma	Beta	Phi
-.781	-.016	-35.000
-.506	-.016	-25.000
-.280	-.016	-15.000
-.178	-.017	-10.000
-.092	-.018	-5.000
-.048	-.018	-2.500
-.004	-.018	0.000
.032	-.018	2.500
.061	-.018	5.000
.130	-.019	10.000
.190	-.019	15.000
.346	-.020	25.000
.583	-.018	35.000

3. Mach number

Gamma	Beta	Phi
-.787	-.024	-35.000
-.510	-.024	-25.000
-.276	-.024	-15.000
-.172	-.026	-10.000
-.085	-.027	-5.000
-.050	-.027	-2.500
-.003	-.027	0.000
.037	-.027	2.500
.066	-.027	5.000
.131	-.027	10.000
.191	-.029	15.000
.345	-.029	25.000
.592	-.027	35.000

4. Mach number

Gamma	Beta	Phi
-.790	-.035	-35.000
-.519	-.035	-25.000
-.276	-.035	-15.000
-.174	-.037	-10.000
-.055	-.039	-5.000
-.043	-.039	-2.500
-.003	-.038	0.000
.037	-.038	2.500
.064	-.039	5.000
.133	-.039	10.000
.196	-.041	15.000
.349	-.042	25.000
.596	-.039	35.000

5. Mach number

Gamma	Beta	Phi
-.795	-.045	-35.000
-.521	-.045	-25.000
-.282	-.045	-15.000
-.175	-.048	-10.000
-.082	-.050	-5.000
-.040	-.050	-2.500
-.010	-.050	0.000
.034	-.049	2.500
.067	-.049	5.000
.134	-.050	10.000
.198	-.053	15.000
.350	-.055	25.000
.593	-.050	35.000

6. Mach number

Gamma	Beta	Phi
-.803	-.065	-35.000
-.528	-.065	-25.000
-.284	-.065	-15.000
-.173	-.068	-10.000
-.085	-.071	-5.000
-.032	-.072	-2.500
-.004	-.071	0.000
.024	-.071	2.500
.058	-.071	5.000
.131	-.071	10.000
.199	-.075	15.000
.349	-.078	25.000
.603	-.073	35.000

X1	X2	Y
----	----	---

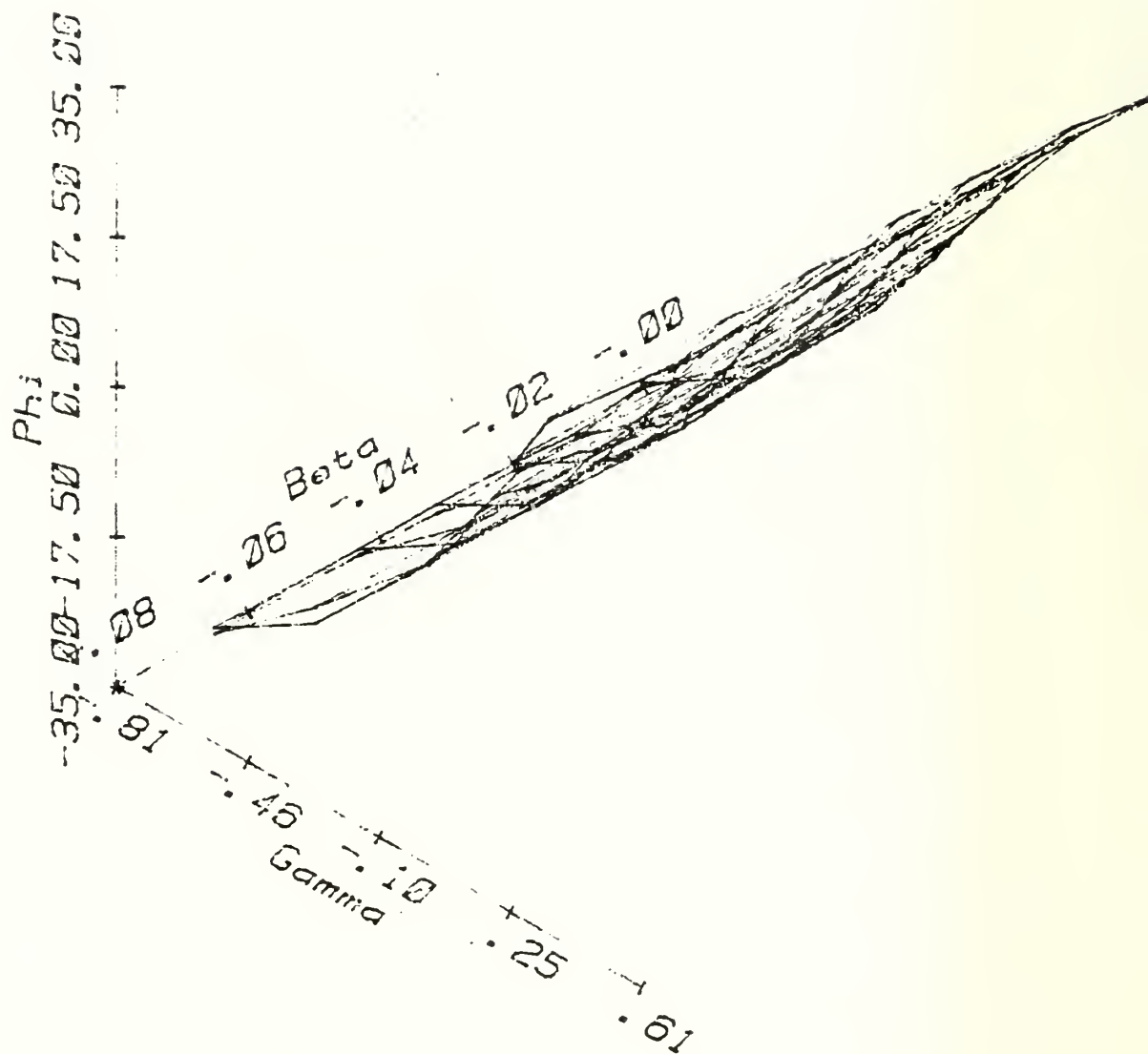
System matrix A and vector B before Gauss Jordan Elimination

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1.00E+00	-.272E-04	-.777E-05	.933E+01	-.312E+00	.145E-01							
2	-.272E-04	.777E-05	-.890E-06	-.312E+00	.145E-01	-.790E-03							
3	-.777E-05	-.890E-06	.852E-07	.145E-01	-.790E-03	.471E-04							
4	.933E+01	-.312E+00	.145E-01	-.790E-03	.471E-04	-.298E-05							
5	-.312E+00	.145E-01	-.790E-03	.471E-04	-.298E-05	.195E-06							
6	.145E-01	-.790E-03	.471E-04	-.298E-05	.195E-06	-.131E-07							
7	-.790E-03	.471E-04	-.298E-05	.195E-06	-.131E-07	.600E-05							
8	.471E-04	-.298E-05	.195E-06	-.131E-07	.600E-05	-.274E-06							
9	-.298E-05	.195E-06	-.131E-07	.600E-05	-.274E-06	.122E+00							
10	.195E-06	-.131E-07	.600E-05	-.274E-06	.122E+00	.561E-02							
11	-.131E-07	.600E-05	-.274E-06	.122E+00	.561E-02	-.300E-03							
12	.600E-05	-.274E-06	.122E+00	.561E-02	-.300E-03	.176E-04							
13	-.274E-06	.122E+00	.561E-02	-.300E-03	.176E-04	-.108E-05							
14	.122E+00	.561E-02	-.300E-03	.176E-04	-.108E-05	.694E-07							
15	.561E-02	-.300E-03	.176E-04	-.108E-05	.694E-07	-.219E-02							
16	-.300E-03	.176E-04	-.108E-05	.694E-07	-.219E-02	.108E-03							
17	.176E-04	-.108E-05	.694E-07	-.219E-02	.108E-03	-.578E-05							
18	-.108E-05	.694E-07	-.219E-02	.108E-03	-.578E-05	.320E-06							
19	.694E-07	-.219E-02	.108E-03	-.578E-05	.320E-06	-.181E-07							
20	-.219E-02	.108E-03	-.578E-05	.320E-06	-.181E-07	.000E+00							

Equation system after Gauss Jordan Elimination

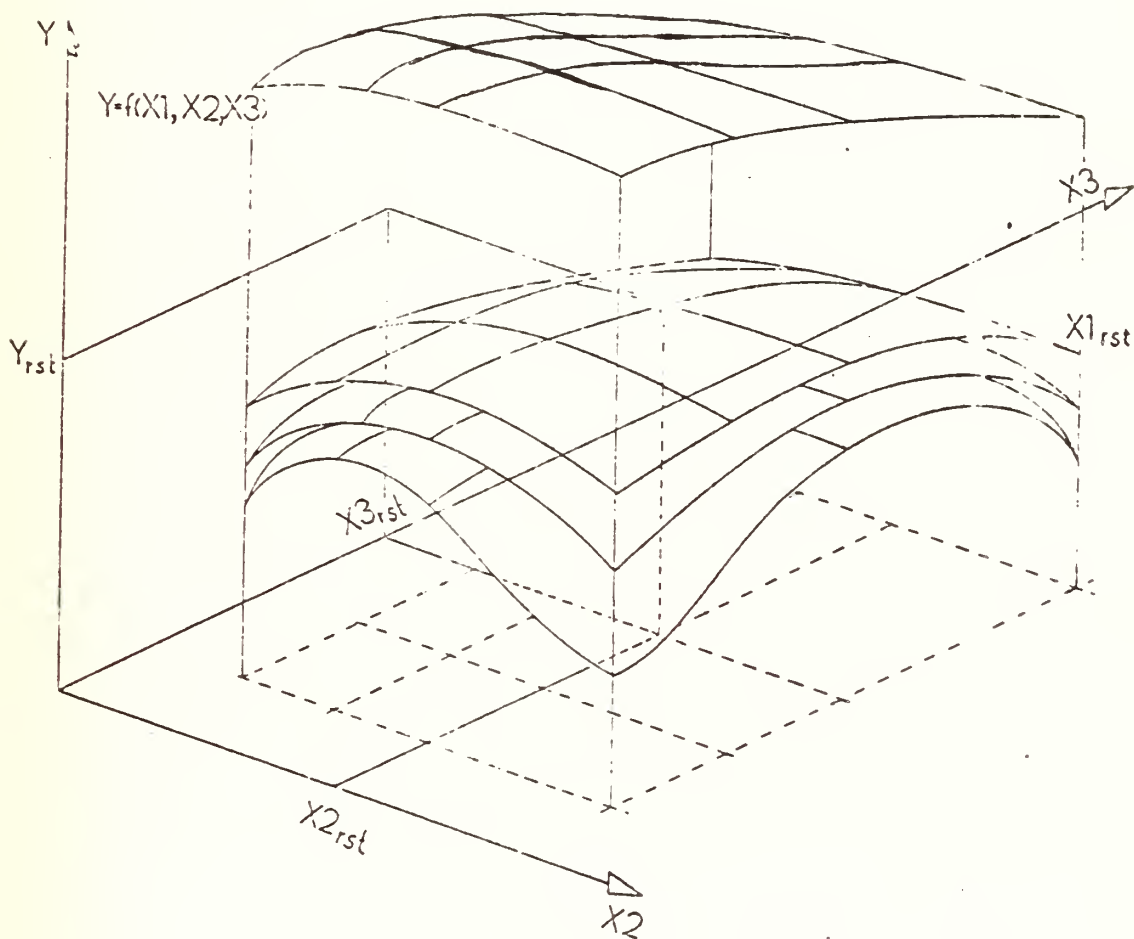
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
3	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
4	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
5	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
6	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
7	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
8	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
9	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00	.000E+00
10	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00	.000E+00
11	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00	.000E+00
12	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	1.00E+00	.000E+00

ii) graphic output



4. FOUR DIMENSIONAL APPROXIMATION

4.1. Problem:



A data set of $NPNTS1 * NPNTS2 * NPNTS3$ data points is given, where Y depends on the parameters $X1$, $X2$ and $X3$. The data pattern is to be approximated by a function $Y = f(X1, X2, X3)$, so that the error between data points and analytically determined points is lower.

4.2. Approach

Again, as in the two and three dimensional case, a polynomial method is used. For

$$Y = f(X_1, X_2, X_3)$$

we assume

$$\begin{aligned}
 Y = & \{ [C_{11} + C_{12}X_3 + \dots + C_{1N}X_3^{N-1}] + \\
 & + [C_{21} + C_{22}X_3 + \dots + C_{2N}X_3^{N-1}] \cdot X_2 + \\
 & \dots \\
 & + [C_{M1} + C_{M2}X_3 + \dots + C_{MN}X_3^{N-1}] \cdot X_2^{M-1} \} + \\
 & + \{ [C_{211} + C_{212}X_3 + \dots + C_{21N}X_3^{N-1}] + \\
 & [C_{221} + C_{222}X_3 + \dots + C_{22N}X_3^{N-1}] \cdot X_2 + \\
 & \dots \\
 & + [C_{2M1} + C_{2M2}X_3 + \dots + C_{2MN}X_3^{N-1}] \cdot X_2^{M-1} \} X_1 + \\
 & \dots \\
 & + \{ [C_{L11} + C_{L12}X_3 + \dots + C_{L1N}X_3^{N-1}] + \\
 & + [C_{L21} + C_{L22}X_3 + \dots + C_{L2N}X_3^{N-1}] \cdot X_2 + \\
 & \dots \\
 & + [C_{LM1} + C_{LM2}X_3 + \dots + C_{LMN}X_3^{N-1}] \cdot X_2^{M-1} \} X_1^{L-1}
 \end{aligned}$$

or

$$Y = \sum_{i=1}^L \left\{ \sum_{j=1}^M \left\{ \sum_{k=1}^N C_{ijk} X_3^{k-1} \right\} X_2^{j-1} \right\} X_1^{i-1} \quad (4.1.)$$

This data arrangement which discriminates between data points taken at constant X_1 , X_2 and X_3 , respectively is needed in the application of this method to the description of pneumatic-velocity probe characteristics. The various surfaces in the above sketch for example can be seen as data taken at different Mach number (X_1). At each constant Mach number ($X_1 = \text{constant}$) yaw angle (X_2) and pitch angle (X_3) are varied, and the probe is measured to be Y . Even though probes are usually balanced to the average flow yaw angle in some cases this is not possible and the four dimensional approximation is then needed. A similar method for describing probe calibration data is used at the institute for Jet Engines and Turbomachines at the Technical University Aachen (Ref. 2).

The least squares criterion is then

$$R = \sum_{r=1}^{NPNTS1} \left\{ \sum_{s=1}^{NPNTS2} \left\{ \sum_{t=1}^{NPNTS3} [f(x1_{rst}, x2_{rst}, x3_{rst}) - Y_{rst}]^2 \right\} \right\} \quad (4.2.)$$

where the indices rst denote data points. Using equation (4.1.), R becomes

$$\begin{aligned} R = \sum_{r=1}^{NPNTS1} \left\{ \sum_{s=1}^{NPNTS2} \left\{ \sum_{t=1}^{NPNTS3} \right. \right. & \left. \left[c_{111} + c_{112} x3_{rst} + \dots + c_{11N} x3_{rst}^{N-1} \right. \right. \\ & + (c_{121} + c_{122} x3_{rst} + \dots + c_{12N} x3_{rst}^{N-1}) \cdot x2_{rst} + \\ & \dots \\ & + (c_{1M1} + c_{1M2} x3_{rst} + \dots + c_{1MN} x3_{rst}^{N-1}) \cdot x2_{rst}^{M-1} + \\ & + (c_{211} + c_{212} x3_{rst} + \dots + c_{21N} x3_{rst}^{N-1} + \\ & + (c_{221} + c_{222} x3_{rst} + \dots + c_{22N} x3_{rst}^{N-1}) \cdot x2_{rst} + \\ & \dots \\ & + (c_{2M1} + c_{2M2} x3_{rst} + \dots + c_{2MN} x3_{rst}^{N-1}) \cdot x2_{rst}^{M-1} \cdot x1_{rst} \\ & \dots \end{aligned}$$

$$\begin{aligned}
& + \langle C_{L11} + C_{L12} X_{rst}^3 + \dots + C_{L1N} X_{rst}^{N-1} + \\
& (C_{L21} + C_{L22} X_{rst}^3 + \dots + C_{L2N} X_{rst}^{N-1}) \cdot X_{rst}^2 + \\
& \dots \\
& (C_{LM1} + C_{LM2} X_{rst}^3 + \dots + C_{LMN} X_{rst}^{N-1}) \cdot X_{rst}^{M-1} \cdot X_{rst}^{L-1} \rangle \} \}
\end{aligned}$$

The term in the [] - bracket shall be called B

$$R = \sum_{r=1}^{NPNTS1} \{ \sum_{s=1}^{NPNTS2} \{ \sum_{t=1}^{NPNTS3} B^2 \} \}$$

4.3. Solution:

To minimize the error, R is partially differentiated with respect to the coefficients C_{ijk} and the partial derivatives are set to zero, thus:

$$\frac{\partial R}{\partial C_{ijk}} \stackrel{!}{=} 0 \quad i=1 \dots L; j=1 \dots M; k=1 \dots N$$

$$\frac{\partial R}{\partial C_{ijk}} = \sum_{r=1}^{NPNTS1} \left\{ \sum_{s=1}^{NPNTS2} \left\{ \sum_{t=1}^{NPNTS3} 2B \cdot \frac{\partial B}{\partial C_{ijk}} \right\} \right\} = 0$$

Assuming, that the summations extend over all data points, $\Sigma\Sigma\Sigma$ should be understood to be $\sum_{r=1}^{NPNTS1} \sum_{s=1}^{NPNTS2} \sum_{t=1}^{NPNTS3}$. Now even a very large sheet of paper can't take the system matrix, unless we skip to submatrix notation.

$$\begin{vmatrix} A_{11}^* & A_{12}^* & \dots & A_{1L}^* \\ A_{21}^* & A_{22}^* & \dots & A_{2L}^* \\ \dots & & & \\ A_{L1}^* & A_{L2}^* & \dots & A_{LL}^* \end{vmatrix} \begin{vmatrix} C_1^* \\ C_2^* \\ \dots \\ C_L^* \end{vmatrix} = \begin{vmatrix} B_1^* \\ B_2^* \\ \dots \\ B_L^* \end{vmatrix} \quad (4.3.)$$

The asterisk indicates a submatrix 1. class. On the following pages the submatrices and subvectors 1. class are listed.

[illegible]

C_{111}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_1^* =$	$\mathcal{B}_1^* =$
C_{112}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{11N}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
C_{121}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_2^* =$	$\mathcal{B}_2^* =$
C_{122}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{12N}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
C_{1N1}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_3^* =$	$\mathcal{B}_3^* =$
C_{1N2}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{1NN}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		

C_{211}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_4^* =$	$\mathcal{B}_4^* =$
C_{212}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{21N}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
C_{221}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_5^* =$	$\mathcal{B}_5^* =$
C_{222}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{22N}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
C_{2N1}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_6^* =$	$\mathcal{B}_6^* =$
C_{2N2}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{2NN}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		

C_{311}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_7^* =$	$\mathcal{B}_7^* =$
C_{312}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{31N}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
C_{321}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_8^* =$	$\mathcal{B}_8^* =$
C_{322}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{32N}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
C_{3N1}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$	$\mathcal{C}_9^* =$	$\mathcal{B}_9^* =$
C_{3N2}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		
\dots	\dots		
C_{3NN}	$\sum \sum \sum Y_{1st} X_{1st}^{1st} X_{2st}^{2nd} X_{3st}^{3rd}$		

4.4. Structure of Equation System:

4.4.1. System Matrix A

It can be seen, that all submatrices 1. class on the same diagonal bands 2. order of the system matrix are identical and therefore will be renamed from $A_{i,j}^*$ to A_k^*

$$A_1^* = A_{1,1}^*$$

$$A_2^* = A_{2,1}^* = A_{1,2}^*$$

$$A_3^* = A_{3,1}^* = A_{2,2}^* = A_{1,3}^*$$

...

$$A_L^* = A_{L,1}^* = A_{L-1,2}^* = A_{L-2,3}^* = \dots = A_{2,L-1}^* = A_{1,L}^*$$

$$A_{L+1}^* = A_{L,2}^* = A_{L-1,3}^* = \dots = A_{3,L-1}^* = A_{2,L}^*$$

$$A_{L+2}^* = A_{L,3}^* = \dots = A_{4,L-1}^* = A_{3,L}^*$$

...

$$A_{2L-1}^* = A_{L,L}^*$$

The submatrices 1. class A_k^* ($k=1, \dots, 2L-1$) themselves are divided up into submatrices 2. class.

$$A_k^* = \begin{vmatrix} A_{k;1,1}^{**} & A_{k;1,2}^{**} & \dots & A_{k;1,M}^{**} \\ A_{k;2,1}^{**} & A_{k;2,2}^{**} & \dots & A_{k;2,M}^{**} \\ \dots & \dots & \dots & \dots \\ A_{k;M,1}^{**} & A_{k;M,2}^{**} & \dots & A_{k;M,M}^{**} \end{vmatrix}$$

The double asterisk indicates a submatrix 2. class. All submatrices

2. class on the same diagonal bands 2. order of the submatrices 1. class

A_k^* ($k = 1, \dots, 2L$) are identical and therefore will be renamed from

$A_{k;i,j}^{**}$ to $A_{k;1}^{**}$

$$A_{k;1}^{**} = A_{k;1,1}^{**}$$

$$A_{k;2}^{**} = A_{k;2,1}^{**} = A_{k;1,2}^{**}$$

$$A_{k;3}^{**} = A_{k;3,1}^{**} = A_{k;2,2}^{**} = A_{k;1,3}^{**}$$

...

$$A_{k;M}^{**} = A_{k;M,1}^{**} = A_{k;M-1,2}^{**} = A_{k;M-2,3}^{**} = \dots = A_{k;2,M-1}^{**} = A_{k;1,M}^{**}$$

$$A_{k;M+1}^{**} = A_{k;M,2}^{**} = A_{k;M-1,3}^{**} = \dots = A_{k;3,M-1}^{**} = A_{k;2,M}^{**}$$

$$A_{k;M+2}^{**} = A_{k;M,3}^{**} = \dots = A_{k;4,M-1}^{**} = A_{k;3,M}^{**}$$

...

$$A_{k;2M-1}^{**} = A_{k;1,M}^{**}$$

where $A_{k;1}^{**}$ can be written as

$$A_{k;1}^{**} = \begin{vmatrix} \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} & \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^2 & \dots & \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^{N-1} \\ \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^2 & \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^3 & \dots & \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^N \\ \dots & \dots & \dots & \dots \\ \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^{M-1} & \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^N & \dots & \sum \sum \sum x_{nt}^{k-1} \cdot x_{nt}^{l-1} \cdot x_{nt}^{2N-2} \end{vmatrix}$$

$$k = 1, \dots, 2L-1; l = 1, \dots, 2M-1$$

Note, that this submatrix 2. class is not only symmetrical, but also the elements on each diagonal line 2. order are identical. The elements $a_{k;l;i,j}$ of the submatrix 2. class $A_{k;l}^{**}$ therefore are renamed to $a_{k;l;m}$; L specifies the place of the submatrix 2. class $A_{k;l}^{**}$ in the submatrix 1. class A_k^* , k specifies the place of the submatrix 1. class A_k^* in the system matrix A and i, j specifies the place of the element $a_{k;l;i,j}$ in the submatrix 2. class $A_{k;l}^{**}$

$$a_{k;l;1} = a_{k;l;1,1}$$

$$a_{k;l;2} = a_{k;l;2,1} = a_{k;l;1,2}$$

$$a_{k;l;3} = a_{k;l;3,1} = a_{k;l;2,2} = a_{k;l;1,3}$$

...

$$a_{k;l;N} = a_{k;l;N,1} = a_{k;l;N-1,2} = a_{k;l;N-2,3} = \dots = a_{k;l;2,N-1} = a_{k;l;1,N}$$

$$a_{k;l;N+1} = a_{k;l;N,2} = a_{k;l;N-1,3} = \dots = a_{k;l;3,N-1} = a_{k;l;2,N}$$

$$a_{k;l;N+2} = a_{k;l;N,3} = \dots = a_{k;l;4,N-1} = a_{k;l;3,N}$$

...

$$a_{k;l;2N-1} = a_{k;l;N,N}$$

where $a_{k;l,m}$ can be written as

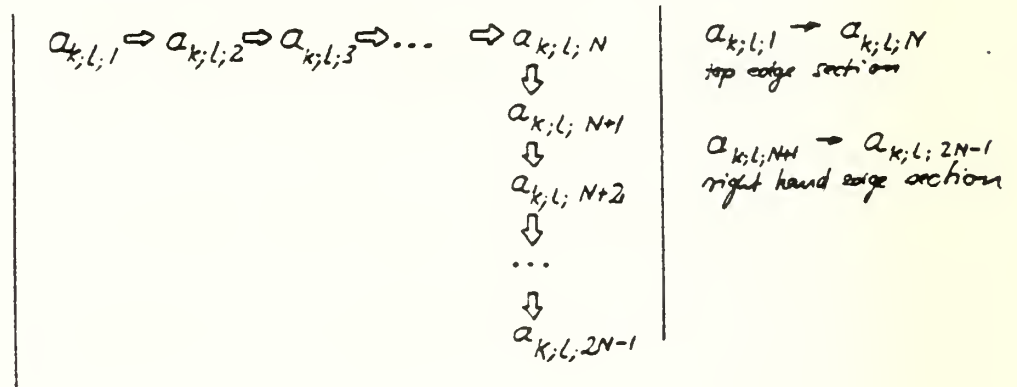
$$a_{k;l,m} = \sum_{r=1}^{NPM1} \left\{ \sum_{s=1}^{NPM2} \left\{ \sum_{t=1}^{NPM3} X1_{rst}^{k-1} \cdot X2_{rst}^{L-1} \cdot X3_{rst}^{m-1} \right\} \right\} \quad (4.4.)$$

$$k = 1, \dots, 2L-1; \quad L = 1, \dots, 2M-1; \quad m = 1, \dots, 2N-1$$

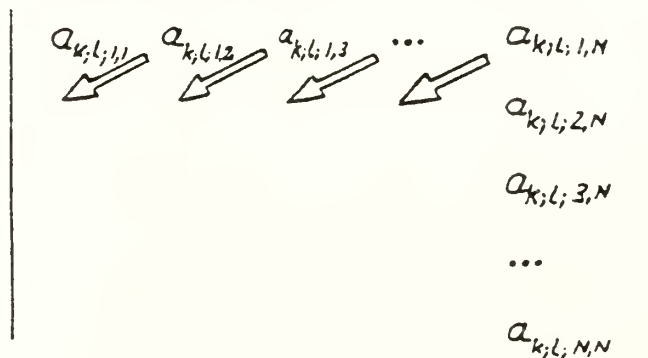
This operation is programmed in REAL FUNCTION S4 (NPNTS1, NPNTS2, NPNTS3, IPOWR1, IPOWR2, IPOWR3, IY). The data X1, X2, X3 and Y are known to this function through a common block named DATA4. So only NPNTS1, NPNTS2, NPNTS3, IPOWR1 (= k-1), IPOWR2 (= L-1), IPOWR3 (= m-1) and IY (=0) have to be passed to the function and the value of $a_{k;l;m}$ is returned through the function name S4.

SUBROUTINE MAT42 presets the submatrices 2. class $A_{k,l}^{**}$ in the following way

- i) Preset edge section elements (using REAL FUNCTION S4)



- ii) Copy defined elements diagonally



$$\begin{array}{cccccc}
 a_{k,l;1,1} & a_{k,l;1,2} & a_{k,l;1,3} & \dots & a_{k,l;1,N} \\
 \swarrow & \swarrow & \swarrow & \swarrow & \\
 a_{k,l;2,1} & a_{k,l;2,2} & a_{k,l;2,3} & \dots & a_{k,l;2,N} \\
 & & & & a_{k,l;3,N} \\
 & & & & \dots \\
 & & & & a_{k,l;N,N}
 \end{array}$$

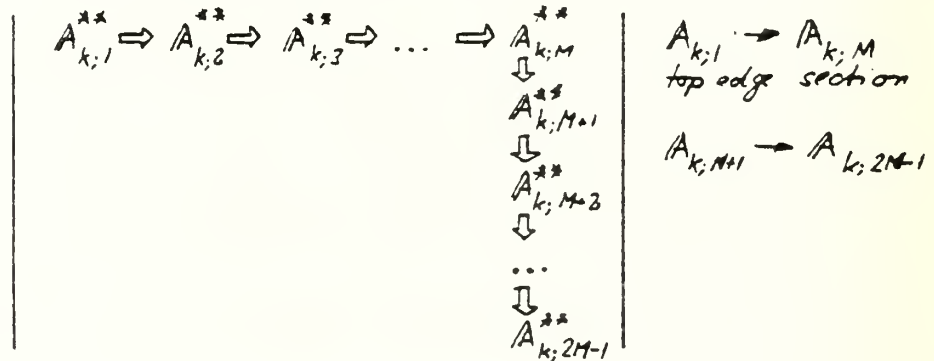
$$\begin{array}{cccccc}
 a_{k,l;1,1} & a_{k,l;1,2} & a_{k,l;1,3} & \dots & a_{k,l;1,N} \\
 a_{k,l;2,1} & a_{k,l;2,2} & a_{k,l;2,3} & \dots & a_{k,l;2,N} \\
 \swarrow & \swarrow & \swarrow & \swarrow & \\
 a_{k,l;3,1} & a_{k,l;3,2} & a_{k,l;3,3} & \dots & a_{k,l;3,N} \\
 & & & & \dots \\
 & & & & a_{k,l;N,N}
 \end{array}$$

$$\begin{array}{cccccc}
 a_{k,l;1,1} & a_{k,l;1,2} & a_{k,l;1,3} & \dots & a_{k,l;1,N} \\
 a_{k,l;2,1} & a_{k,l;2,2} & a_{k,l;2,3} & \dots & a_{k,l;2,N} \\
 a_{k,l;3,1} & a_{k,l;3,2} & a_{k,l;3,3} & \dots & a_{k,l;3,N} \\
 \dots & \dots & \dots & \dots & \dots \\
 \swarrow & \swarrow & \swarrow & \swarrow & \\
 & & & & a_{k,l;N,N}
 \end{array}$$

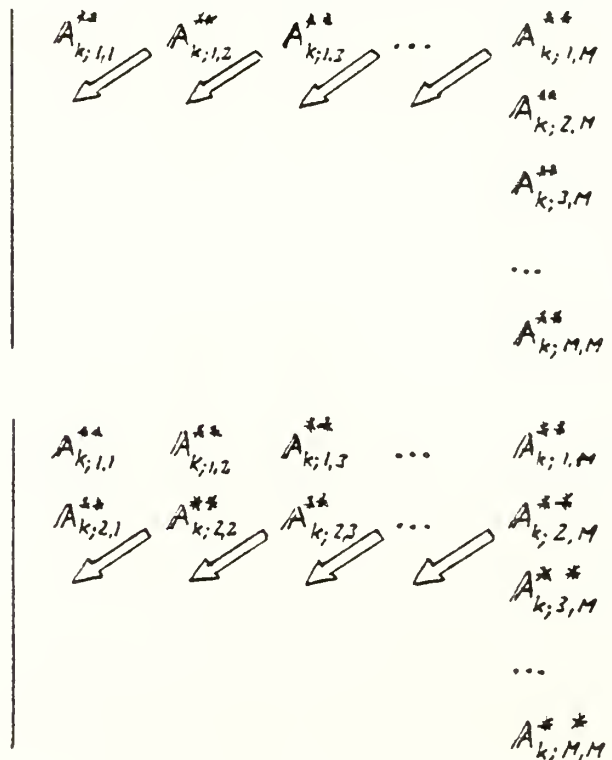
$$A_{k,l}^{**} = \begin{array}{cccccc}
 a_{k,l;1,1} & a_{k,l;1,2} & a_{k,l;1,3} & \dots & a_{k,l;1,N} \\
 a_{k,l;2,1} & a_{k,l;2,2} & a_{k,l;2,3} & \dots & a_{k,l;2,N} \\
 a_{k,l;3,1} & a_{k,l;3,2} & a_{k,l;3,3} & \dots & a_{k,l;3,N} \\
 \dots & \dots & \dots & \dots & \dots \\
 a_{k,l;N,1} & a_{k,l;N,2} & a_{k,l;N,3} & \dots & a_{k,l;N,N}
 \end{array}$$

This subroutine that needs NPNTS1, NPNTS2, NPNTS3, IPOWR1(=k-1) and IPOWR2(=L-1) as input parameters, returns the submatrix 2. class $A_{k;L}^{**}$ (SUBM2(4,4)) to SUBROUTINE MAT41, which foresets the submatrices 1. class A_k^* in the following way

i) Preset edge section submatrices (using SUBROUTINE MAT42)



ii) Copy defined submatrices diagonally



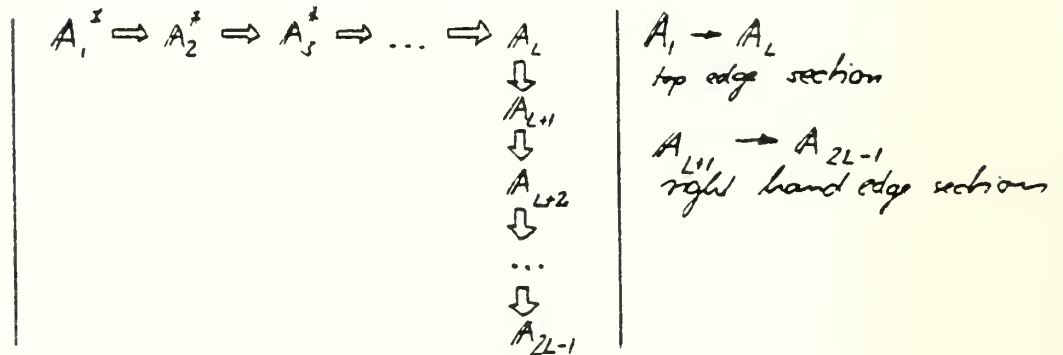
$$\begin{array}{ccccc}
 A_{k,1,1}^{**} & A_{k,1,2}^{**} & A_{k,1,3}^{**} & \dots & A_{k,1,M}^{**} \\
 A_{k,2,1}^{**} & A_{k,2,2}^{**} & A_{k,2,3}^{**} & \dots & A_{k,2,M}^{**} \\
 A_{k,3,1}^{**} & A_{k,3,2}^{**} & A_{k,3,3}^{**} & \dots & A_{k,3,M}^{**} \\
 & \swarrow & \swarrow & \swarrow & \swarrow \\
 & & & & \dots \\
 & & & & A_{k,M,M}^{**}
 \end{array}$$

$$\begin{array}{ccccc}
 A_{k,1,1}^{**} & A_{k,1,2}^{**} & A_{k,1,3}^{**} & \dots & A_{k,1,M}^{**} \\
 A_{k,2,1}^{**} & A_{k,2,2}^{**} & A_{k,2,3}^{**} & \dots & A_{k,2,M}^{**} \\
 A_{k,3,1}^{**} & A_{k,3,2}^{**} & A_{k,3,3}^{**} & \dots & A_{k,3,M}^{**} \\
 \dots & \dots & \dots & \dots & \dots \\
 \swarrow & \swarrow & \swarrow & \swarrow & \swarrow \\
 & & & & A_{k,M,M}^{**}
 \end{array}$$

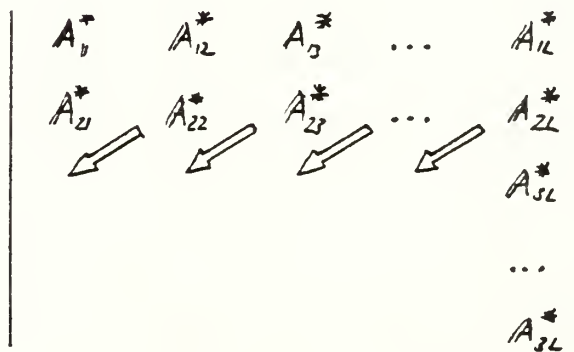
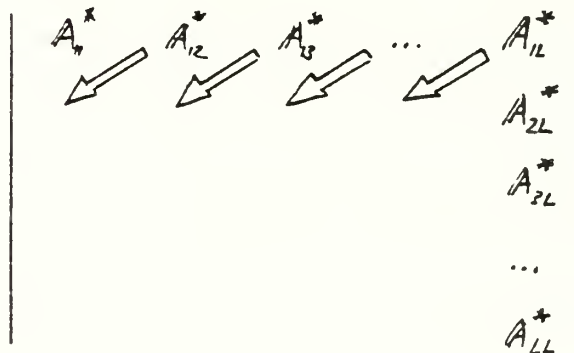
$$A_k^* = \begin{array}{ccccc}
 A_{k,1,1}^{**} & A_{k,1,2}^{**} & A_{k,1,3}^{**} & \dots & A_{k,1,M}^{**} \\
 A_{k,2,1}^{**} & A_{k,2,2}^{**} & A_{k,2,3}^{**} & \dots & A_{k,2,M}^{**} \\
 A_{k,3,1}^{**} & A_{k,3,2}^{**} & A_{k,3,3}^{**} & \dots & A_{k,3,M}^{**} \\
 \dots & \dots & \dots & \dots & \dots \\
 A_{k,M,1}^{**} & A_{k,M,2}^{**} & A_{k,M,3}^{**} & \dots & A_{k,M,M}^{**}
 \end{array}$$

This subroutine, that needs NPNTS1, NPNTS2, NPNTS3, M, N, IPOWR1 (=L-1) as input parameters, returns the subroutine 1st class A_k^* (SUBM1(16,16)) to SUBROUTINE MAT4, which presets the entire system matrix A in the following way.

i) Preset edge section submatrices (using SUBROUTINE MAT41)



ii) Copy defined submatrices diagonally



$$\begin{array}{ccccc}
 A_{11}^* & A_{12}^* & A_{13}^* & \dots & A_{1L}^* \\
 A_{21}^* & A_{22}^* & A_{23}^* & \dots & A_{2L}^* \\
 A_{31}^* & A_{32}^* & A_{33}^* & \dots & A_{3L}^* \\
 \swarrow & \swarrow & \swarrow & \swarrow & \dots \\
 & & & & A_{LL}
 \end{array}$$

$$\begin{array}{ccccc}
 A_{11}^* & A_{12}^* & A_{13}^* & \dots & A_{1L}^* \\
 A_{21}^* & A_{22}^* & A_{23}^* & \dots & A_{2L}^* \\
 A_{31}^* & A_{32}^* & A_{33}^* & \dots & A_{3L}^* \\
 \dots & \dots & \dots & \dots & \dots \\
 \swarrow & \swarrow & \swarrow & \swarrow & \dots
 \end{array}$$

$$A = \begin{array}{ccccc}
 A_{11}^* & A_{12}^* & A_{13}^* & \dots & A_{1L}^* \\
 A_{21}^* & A_{22}^* & A_{23}^* & \dots & A_{2L}^* \\
 A_{31}^* & A_{32}^* & A_{33}^* & \dots & A_{3L}^* \\
 \dots & \dots & \dots & \dots & \dots \\
 A_{L1} & A_{L2} & A_{L3} & \dots & A_{LL}
 \end{array}$$

4.4.2. Right Hand Side Vector B:

The right hand side subvectors 1. class B_k^* generally can be written as

$$B_k^* =$$

$$\begin{array}{|l}
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X3_{nst} \\
 \dots \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X3_{nst}^{N-1} \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X2_{nst} \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X2_{nst} \cdot X3_{nst} \\
 \dots \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X2_{nst} \cdot X3_{nst}^{N-1} \\
 \\
 \dots \\
 \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X2_{nst}^{M-1} \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X2_{nst}^{M-1} \cdot X3_{nst} \\
 \dots \\
 \sum \sum \sum Y_{nst} \cdot X1_{nst}^{k-1} \cdot X2_{nst}^{M-1} \cdot X3_{nst}^{N-1}
 \end{array}$$

$k = 1, \dots, L$

The subvectors 1. class B_k^* ($k = 1, \dots, L$) themselves will be divided up into subvectors 2. class:

$$B_k^* = \begin{vmatrix} B_{k,1}^{**} \\ B_{k,2}^{**} \\ \dots \\ B_{k,L}^{**} \end{vmatrix}$$

where subvectors 2. class $B_{k;L}^{**}$ can be written as

$$B_{k;L}^{**} = \begin{vmatrix} \sum \sum \sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{L-1} \\ \sum \sum \sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{L-1} \cdot X3 \\ \sum \sum \sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{L-1} \cdot X3^{N-1} \end{vmatrix}$$

$$k = 1, \dots, L; \quad L = 1, \dots, M; \quad m = 1, \dots, N$$

Finally, the elements of the subvectors 2. class $B_{k;1}$ can be written as

$$b_{k;l;m} = \sum \sum \sum Y_{rst} \cdot X1_{rst}^{k-1} \cdot X2_{rst}^{L-1} \cdot X3_{rst}^{m-1}$$

$$k = 1, \dots, L; \quad L = 1, \dots, M; \quad m = 1, \dots, N$$

To calculate $b_{k;l;m}$, REAL FUNCTION S4 is used, again. Data X1, X2, X3 and Y are available through COMMON block DATA4. NPNTS1, NPNTS2, NPNTS3, IPOWR1 (= k-1), IPOWR2 (= L-1) IPOWR3 (= m-1) and IY (=1) have to be passed to the function and the value of $b_{k;l;m}$ is returned through the function name S4. Using FORTRAN programming language, the allocation of $b_{k;l;m}$ can be performed by three stacked DO-loops.

```
I=0
DO 07 L1=1,L,1
IPOWER1=L1-1
DO 08 M1=1,M,1
IPOWER2=M1-1
DO 08 N1=1,N,1
IPOWER3=N1-1
I=I+1
08 B(I)=S4(NPNTS1,NPNTS2,NPNTS3,IPOWER1,IPOWER2,IPOWER3,1)
```

4.5. Software:

The software to compute the coefficients for a two dimensional approximation is described in APPENDIX A and is implemented in the TPL HP=21MX computer system. To work correctly with these program modules, the user has to conform to the following conventions.

- i) Provide the data in four arrays
(Type: REAL) of 5*5*5* elements
through a COMMON block, named DATA4.

COMMON / DATA4 / X1,X2,X3,Y
REAL X1(5,5,5),X2(5,5,5),X3(5,5,5),Y(5,5,5)
- ii) Dimension an array (Type: REAL) of
5*5*5 elements to contain the coefficients,

REAL COEF (5,5,5)
- iii) Define the parameters NPNTS1, NPNTS2,
NPNTS3,L,M,N and IPRINT

NPNTS1... # of X1 variations
L ≤ NPNTS1 ≤ 5

NPNTS2... # of X2 variations
M ≤ NPNTS2 ≤ 5

NPNTS3... # of X3 variations
N ≤ NPNTS3 ≤ 5

L ... (desired order of approximation
polynomial w.r.t. X1) + 1
1 ≤ L ≤ 4

M ... (desired order of approximation
polynomial w.r.t. X2) + 1
1 ≤ M ≤ 4

N ... (desired order of approximation
polynomial w.r.t. X3) + 1
1 ≤ N ≤ 4

```

IPRINT ... Controls quantity of print out
2 ... Print system matrix and right
      hand side vector before and
      after Gauss Jordan Elimination.
1 ... Print equation system after
      Gauss Jordan Elimination
<0... No print out
>0... Print equation (4.1.) with
      the actual parameters.

```

- iv) If a user program uses subroutine MAT4, the software modules have to be loaded using the procedure in section 1.4.iv.

If all these requirements are met, the correct call for the subroutine is:

```
CALL MAT4 (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT)
```

Upon completed execution of this approximation routine array COEF contains the coefficients. Externals used by MAT4: AB4, DATA4, MAT41, MAT42, S4. Under no circumstances may the user use any of these names for a modules of his own user program.

4.6. Sample User Program

4.6.1 FTN4 Compiler listing of sample user program

```

0001 FTN4,L
0002 PROGRAM FITR (3,99)
0003 C
0004 C APPROXIMATION PROGRAM TO FIT A SURFACE THROUGH DATA POINTS Y,
0005 C WHICH DEPEND ON THREE INDEPENDENT PARAMETERS X1, X2 AND X3.
0006 C
0007 C AUTHOR: HANS ZEBNER
0008 C DATE : AUGUST 26, 1980
0009 C
0010 * , Hans Zebner: 4D-Approximation;  $Y=f(X1,X2,X3)$ .
0011
0012 COMMON / DATA4 / X1,X2,X3,Y
0013 COMMON / AB4 / A,B
0014
0015 REAL X1(5,5,5),X2(5,5,5),X3(5,5,5),Y(5,5,5)
0016 REAL A(64,64),B(64)
0017
0018 REAL COEF(5,5,5)
0019 INTEGER IDC8(144),IFILE(3),NOCR(2),ICLR(3)
0020
0021 DATA NOCR /000033B,040433B/, ICLR /015524B,015515B,006537B/
0022
0023 101 FORMAT (" ENTER NPNTS1 NPNTS2 NPNTS3 "2A2
0024 *)
0025 102 FORMAT (" ENTER L M N "2A2)
0026 103 FORMAT (" ENTER DATA FILE NAME "2A2)
0027 104 FORMAT (3A2)
0028 131 FORMAT (" ENTER IPRINT "2A2)
0029 132 FORMAT (" OUTPUT RAW DATA YES OR NO "2A2)
0030 105 FORMAT ("/" X1")
0031 106 FORMAT ("/X,I2". DATA SET"/)
0032 107 FORMAT (3X,I2I10)
0033 108 FORMAT (1X,I2,14F10.3)
0034 109 FORMAT ("/" X2")
0035 110 FORMAT ("/" X3")
0036 111 FORMAT ("/" Y")
0037 149 FORMAT ("(("3A2)))
0038 1101 FORMAT ("HANNES")
0039 C
0040 C READ RAW DATA FROM FILE
0041 C
0042 01 WRITE (1,101) NOCR
0043 READ (1,*) NPNTS1,NPNTS2,NPNTS3
0044 WRITE (1,149) ICLR
0045 NMAX=5
0046 IF (NPNTS1.GT.NMAX) GO TO 01
0047 IF (NPNTS2.GT.NMAX) GO TO 01
0048 IF (NPNTS3.GT.NMAX) GO TO 01
0049 02 WRITE (1,102) NOCR
0050 READ (1,*) L,M,N
0051 WRITE (1,149) ICLR
0052 IF (L*M*N.GT.64) GO TO 02
0053 03 WRITE (1,103) NOCR
0054 READ (1,104) IFILE
0055 WRITE (1,149) ICLR
0056 31 WRITE (1,131) NOCR
0057 READ (1,*) IPRINT
0058 WRITE (1,149) ICLR
0059 WRITE (1,132) NOCR
0060 READ (1,104) IDUM
0061 WRITE (1,149) ICLR
0062 CALL OPEN (IDC8,IERR,IFILE)
0063 IF (IERR.LT.0) GO TO 03
0064 IL=2*NPNTS1*NPNTS2*NPNTS3
0065 CALL READF (IDC8,IERR,X1,IL,LEN,1)
0066 IF (IERR.LT.0) STOP 0001
0067 CALL READF (IDC8,IERR,X2,IL,LEN,54)
0068 IF (IERR.LT.0) STOP 0002
0069 CALL READF (IDC8,IERR,X3,IL,LEN,107)
0070 IF (IERR.LT.0) STOP 0003
0071 CALL READF (IDC8,IERR,Y,IL,LEN,160)
0072 IF (IERR.LT.0) STOP 0004
0073 CALL CLOSE (IDC8,IERR)
0074 IF (IERR.LT.0) STOP 0005
0075 IF (IDUM.NE.2HYE) GO TO 17
0076 WRITE (6,105)

```

```

0077      DO 04 K1=1, NPNTS1, 1
0078      WRITE (6,106) K1
0079      WRITE (6,107) (K3, K3=1, NPNTS3, 1)
0080      DO 04 K2=1, NPNTS2, 1
0081      04 WRITE (6,108) K2, (X1(K1, K2, K3), K3=1, NPNTS3, 1)
0082      WRITE (6,109)
0083      DO 05 K1=1, NPNTS1, 1
0084      WRITE (6,106) K1
0085      WRITE (6,107) (K3, K3=1, NPNTS3, 1)
0086      DO 05 K2=1, NPNTS2, 1
0087      05 WRITE (6,108) K2, (X2(K1, K2, K3), K3=1, NPNTS3, 1)
0088      WRITE (6,110)
0089      DO 06 K1=1, NPNTS1, 1
0090      WRITE (6,106) K1
0091      WRITE (6,107) (K3, K3=1, NPNTS3, 1)
0092      DO 06 K2=1, NPNTS2, 1
0093      06 WRITE (6,108) K2, (X3(K1, K2, K3), K3=1, NPNTS3, 1)
0094      WRITE (6,111)
0095      DO 07 K1=1, NPNTS1, 1
0096      WRITE (6,106) K1
0097      WRITE (6,107) (K3, K3=1, NPNTS3, 1)
0098      DO 07 K2=1, NPNTS2, 1
0099      07 WRITE (6,108) K2, (Y(K1, K2, K3), K3=1, NPNTS3, 1)
0100      17 CALL MAT4 (NPNTS1, NPNTS2, NPNTS3, L, M, N, COEF, IPRINT)
0101      CALL CODE
0102      WRITE (IFILE, 1101)
0103      CALL OPEN (IDCB, IERR, IFILE, IOPTN, 0, 26, 144)
0104      IF (IERR.LT.0) STOP 0006
0105      CALL WRITE (IDCB, IERR, COEF, 250)
0106      IF (IERR.LT.0) STOP 0007
0107      CALL CLOSE (IDCB, IERR)
0108      IF (IERR.LT.0) STOP 0011
0109      STOP 0077
0110      END

```

, FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01259

COMMON = 00000

4.6.2. Load map of sample user program FITR

FITR 10042 12414 Hans Zebner: 4D-Approximation: $Y=f(X1,X2,X3)$.

to load these program modules, enter (from LOADR): MS, %TPLBL

DATA4	12415	14364	4D-Approximation	/ DATA4 /
MAT4	14365	20251	4D-Approximation	arrange system matrix and vector.
MAT41	20252	20666	4D-Approximation	arrange submatrix 1. class.
MAT42	20667	21066	4D-Approximation	arrange submatrix 2. class.
S4	21067	21370	4D-Approximation	compute summations.
AB4	21371	41570	4D-Approximation	/ AB4 /

LOGLU	41571	41646	92067-16268	REV.1903	790228
READF	41647	42610	92067-16125	REV.1940	790719
OPEN	42611	43106	92067-16125	REV.1903	790215
CLOSE	43107	43316	92067-16125	REV.1903	781229
CLRIO	43317	43325	750701	24998-16001	
QVRD.	43326	43326	92067-16125	REV.1903	780526
\$SMVE	43327	43415	92067-16268	REV.1903	790202
LURQ	43416	44000	92067-16268	REV.1903	790223
.DADS	44001	44110	780818	24998-16001	
.DMP	44111	44256	780818	24998-16001	
.DDI	44257	44557	781021	24998-16001	
SESSN	44560	44575	92067-16125	REV.1903	780413
R/W\$	44576	44734	92067-16125	REV.1903	781214
P.PAS	44735	44763	92067-16125	REV.1903	740801
.DNG	44764	44773	780818	24998-16001	
PAUSE	44774	45074	771122	24998-16001	
\$ALRN	45075	45212	92067-16268	REV.1903	770715
FMTIO	45213	46511	24998-16002	REV.1926	790417
ERRO	46512	46601	771122	24998-16001	
ABS	46602	46610	750701	24998-16001	
.DDE	46611	46622	780818	24998-16001	
.DIN	46623	46630	780818	24998-16001	
.RTOI	46631	46724	780921	24998-16001	
.FPWR	46725	46766	781106	24998-16001	
.SKT	46767	47027	770518	24998-16001	
.FCM	47030	47044	750701	24998-16001	
PAU.E	47045	47045	750701	24998-16001	
ERO.E	47046	47046	750701	24998-16001	
\$OPEN	47047	47223	92067-16125	REV.1903	790103
RW\$UB	47224	47571	92067-16125	REV.1903	781003
RWND\$	47572	47714	92067-16125	REV.1903	780801
FRMTR	47715	53352	24998-16002	REV.1926	790503
FMT.E	53353	53353	24998-16002	REV.1901	781107
REIO	53354	53500	92067-16268	REV.1903	790316
RMPAR	53501	53543	781106	24998-16001	
PNAME	53544	53611	771121	24998-16001	
LUTRU	53612	53720	92067-16268	REV.1903	790223
\$SETP	53721	53745	781106	24998-16001	
.CFER	53746	54023	750701	24998-16001	
.LBT	54024	54054	770518	24998-16001	

20 PAGES RELOCATED 20 PAGES REQ'D NO PAGES EMA NO PAGES MSEG
 LINKS:BP PROGRAM:LB LOAD:TE COMMON:NC
 /LOADR:FITR READY AT 3:53 PM FRI., 26 SEPT, 1980
 /LOADR:\$END

4.6.3. Results

printed output

X1

1. DATA SET

	1	2	3	4	5
1	.100	.100	.100	.100	.100
2	.100	.100	.100	.100	.100
3	.100	.100	.100	.100	.100
4	.100	.100	.100	.100	.100
5	.100	.100	.100	.100	.100

2. DATA SET

	1	2	3	4	5
1	.200	.200	.200	.200	.200
2	.200	.200	.200	.200	.200
3	.200	.200	.200	.200	.200
4	.200	.200	.200	.200	.200
5	.200	.200	.200	.200	.200

3. DATA SET

	1	2	3	4	5
1	.300	.300	.300	.300	.300
2	.300	.300	.300	.300	.300
3	.300	.300	.300	.300	.300
4	.300	.300	.300	.300	.300
5	.300	.300	.300	.300	.300

4. DATA SET

	1	2	3	4	5
1	.400	.400	.400	.400	.400
2	.400	.400	.400	.400	.400
3	.400	.400	.400	.400	.400
4	.400	.400	.400	.400	.400
5	.400	.400	.400	.400	.400

5. DATA SET

	1	2	3	4	5
1	.500	.500	.500	.500	.500
2	.500	.500	.500	.500	.500
3	.500	.500	.500	.500	.500
4	.500	.500	.500	.500	.500
5	.500	.500	.500	.500	.500

X2

1. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

2. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

3. DATA SET

	1	2	3	4	5
1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

4. DATA SET

	1	2	3	4	5
--	---	---	---	---	---

1	-20.000	-20.000	-20.000	-20.000	-20.000
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

5. DATA SET

1	-20.000 ¹	-20.000 ²	-20.000 ³	-20.000 ⁴	-20.000 ⁵
2	-10.000	-10.000	-10.000	-10.000	-10.000
3	0.000	0.000	0.000	0.000	0.000
4	10.000	10.000	10.000	10.000	10.000
5	20.000	20.000	20.000	20.000	20.000

X3

1. DATA SET

1	-20.000 ¹	-10.000 ²	0.000 ³	10.000 ⁴	20.000 ⁵
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

2. DATA SET

1	-20.000 ¹	-10.000 ²	0.000 ³	10.000 ⁴	20.000 ⁵
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

3. DATA SET

1	-20.000 ¹	-10.000 ²	0.000 ³	10.000 ⁴	20.000 ⁵
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

4. DATA SET

1	-20.000 ¹	-10.000 ²	0.000 ³	10.000 ⁴	20.000 ⁵
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

5. DATA SET

1	-20.000 ¹	-10.000 ²	0.000 ³	10.000 ⁴	20.000 ⁵
2	-20.000	-10.000	0.000	10.000	20.000
3	-20.000	-10.000	0.000	10.000	20.000
4	-20.000	-10.000	0.000	10.000	20.000
5	-20.000	-10.000	0.000	10.000	20.000

Y

1. DATA SET

1	.400 ¹	.700 ²	.800 ³	.700 ⁴	.400 ⁵
2	.450	.750	.850	.750	.450
3	.500	.800	.900	.800	.500
4	.450	.750	.850	.750	.450
5	.400	.700	.800	.700	.400

2. DATA SET

1	.463 ¹	.763 ²	.863 ³	.763 ⁴	.463 ⁵
2	.513	.813	.913	.813	.513
3	.563	.863	.963	.863	.563
4	.513	.813	.913	.813	.513

5 .463 .763 .863 .763 .463

3. DATA SET

	1	2	3	4	5
1	.650	.950	1.050	.950	.650
2	.700	1.000	1.100	1.000	.700
3	.750	1.050	1.150	1.050	.750
4	.700	1.000	1.100	1.000	.700
5	.650	.950	1.050	.950	.600

4. DATA SET

	1	2	3	4	5
1	.963	1.263	1.363	1.263	.963
2	1.013	1.313	1.413	1.313	1.013
3	1.063	1.363	1.463	1.363	1.063
4	1.013	1.313	1.413	1.313	1.013
5	.963	1.263	1.363	1.263	.963

5. DATA SET

	1	2	3	4	5
1	1.400	1.700	1.800	1.700	1.400
2	1.450	1.750	1.850	1.750	1.450
3	1.500	1.800	1.900	1.800	1.500
4	1.450	1.750	1.850	1.750	1.450
5	1.400	1.700	1.800	1.700	1.400

Coefficients COEF(I,J,K)

I = 1

J	K = 1	2	3
1	-.418E+04	.965E+03	.691E+02
2	-.134E+01	-.390E-01	.282E-02
3	.121E+02	-.284E+01	-.203E+00

I = 2

J	K = 1	2	3
1	.458E+05	-.106E+05	-.757E+03
2	.143E+02	.417E+00	-.300E-01
3	-.133E+03	.312E+02	.222E+01

I = 3

J	K = 1	2	3
1	-.994E+05	.230E+05	.165E+04
2	-.238E+02	-.696E+00	.499E-01
3	.289E+03	-.678E+02	-.484E+01

5. CONCLUSIONS AND RECOMMENDATIONS

Compared to the probe calibration approach described in Ref 3. the method presented here provides a much simpler way to both calculate the probe calibration surfaces from measured data and to apply the calibration to on-line data reduction. The iteration required in the method of Ref 3 is completely omitted. It is noted however, that the Gauss Jordan Elimination lacks some sophistication. The system matrix for example undergoes the elimination without prior conditioning. Particularly in the case of the 3-D and the 4-D approximations, numerical round off errors have an influence on the accuracy of the coefficients. Consequently it is recommended, that the Gauss Jordan Elimination routine be revised to use double precision constants and a routine that conditions the system matrix be added. Time constraints made these steps impossible for the author.

Following these refinements, a calibration procedure, similar to McGuire's (ref 3) should be formalised. Since the 4-D- approximation requires large arrays and therefore extensive CP - memory, extended memory access (EMA) is necessary if the HP 21MX computer is used to perform the calculations.

LIST OF REFERENCES

1. McCracken, D. D. and Dorn, W. S.: Numerical Methods and FORTRAN Programming, Third Printing. New York: Wiley & Sons Inc., 1965. Page 262ff.
2. Simon, H.: Anwendung verschiedener Berechnungsverfahren zur Auslegung eines Überschallveridchler-Laufrades, und dessen experimentielle Untersuchung, Dissertation, RWTH Aachen, 1973, Page 134f.

[Application of Various Calculation Methods for the Design of a Supersonic Rotor and Its Experimental Investigation, Dissertation, Technical University Aachen, 1973.]
3. McGuire, A. G.: Pneumatic Velocity Probe Calibration-Users Manual for Data Acquisition and Reduction. Monterey: Naval Postgraduate School, Turbopropulsion Laboratory, Technical Note 80-01, 1980.

APPENDIX A: Some Useful Matrix Conventions and Operations

A1. Submatrix Notation

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1m} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{2m} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{3m} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{4m} \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{nm} \end{vmatrix} \quad \text{or} \quad \begin{vmatrix} A_{11}^* & A_{12}^* & \dots & A_{1m}^* \\ A_{21}^* & A_{22}^* & \dots & A_{2m}^* \\ A_{n1}^* & A_{n2}^* & \dots & A_{nm}^* \end{vmatrix}$$

defined through elements a_{ij} or submatrices A_{ij}^*

A2. Diagonal Lines and Diagonal Bands 1. order

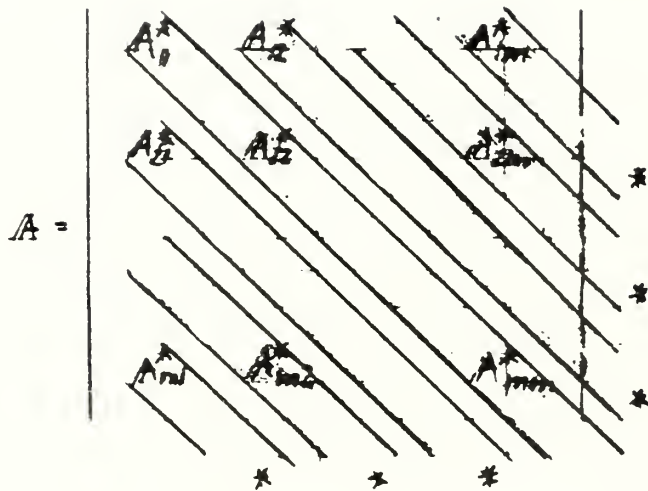
Diagonal lines 1. order and Main diagonal line 1. order

Diagonal bands 1. order and Main diagonal band 1. order

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1m} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{2m} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{3m} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{4m} \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{nm} \end{vmatrix}$$

Main Diagonal Line 1. order
w.r.t. elements

* Diagonal lines 1. order
w.r.t. elements



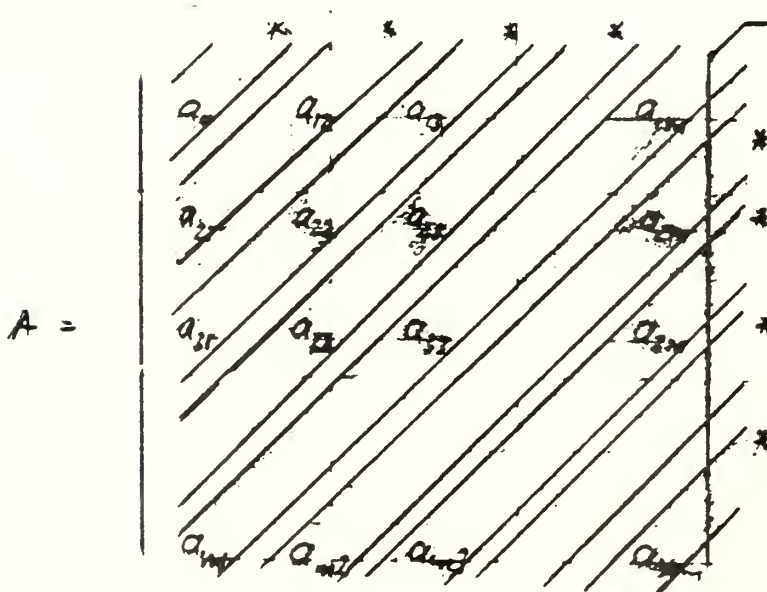
* Diagonal bands 1. order
w.r.t. submatrices

Main Diagonal Band 1. order
w.r.t. Submatrices

A3. Diagonal Lines and Diagonal Bands 2. order

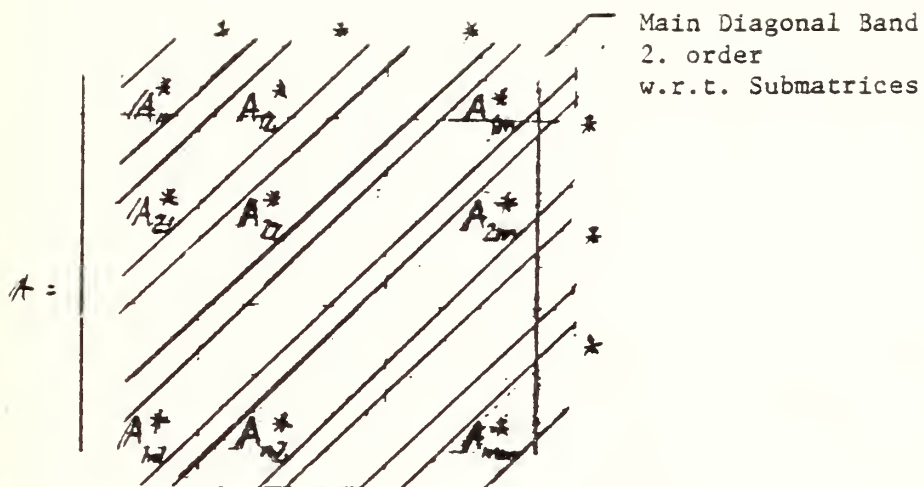
Diagonal lines 2. order and Main diagonal line 2. order

Diagonal bands 2. order and Main diagonal band 2. order



Main Diagonal Line
1. order w.r.t. elements

* Diagonal lines 2. order w.r.t. elements



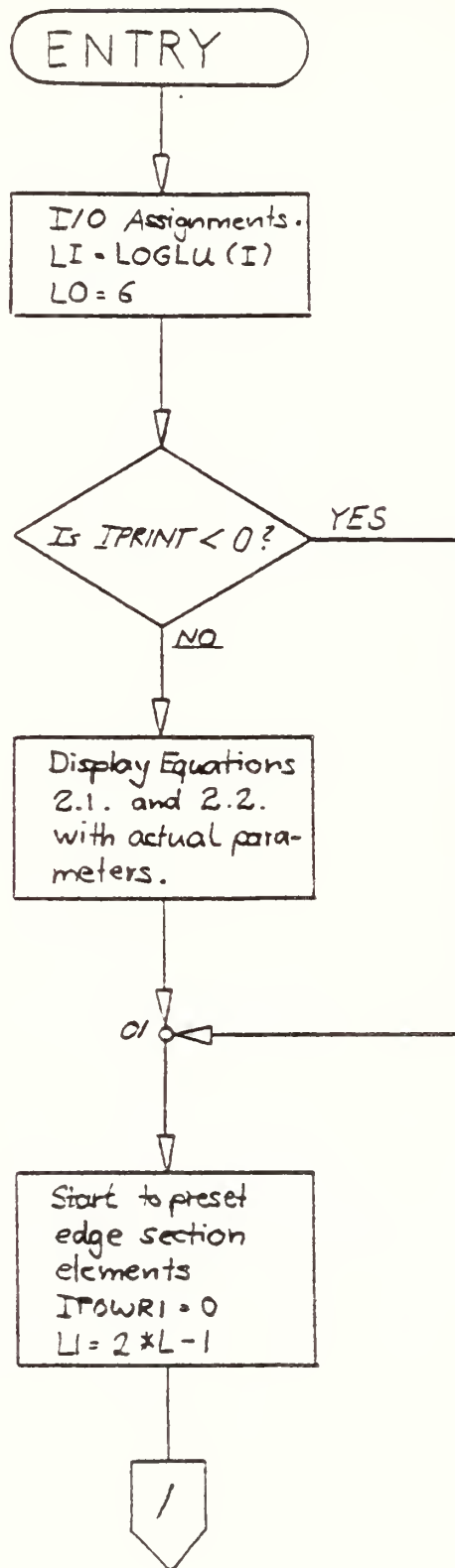
* Diagonal bands 2. order
w.r.t. submatrices

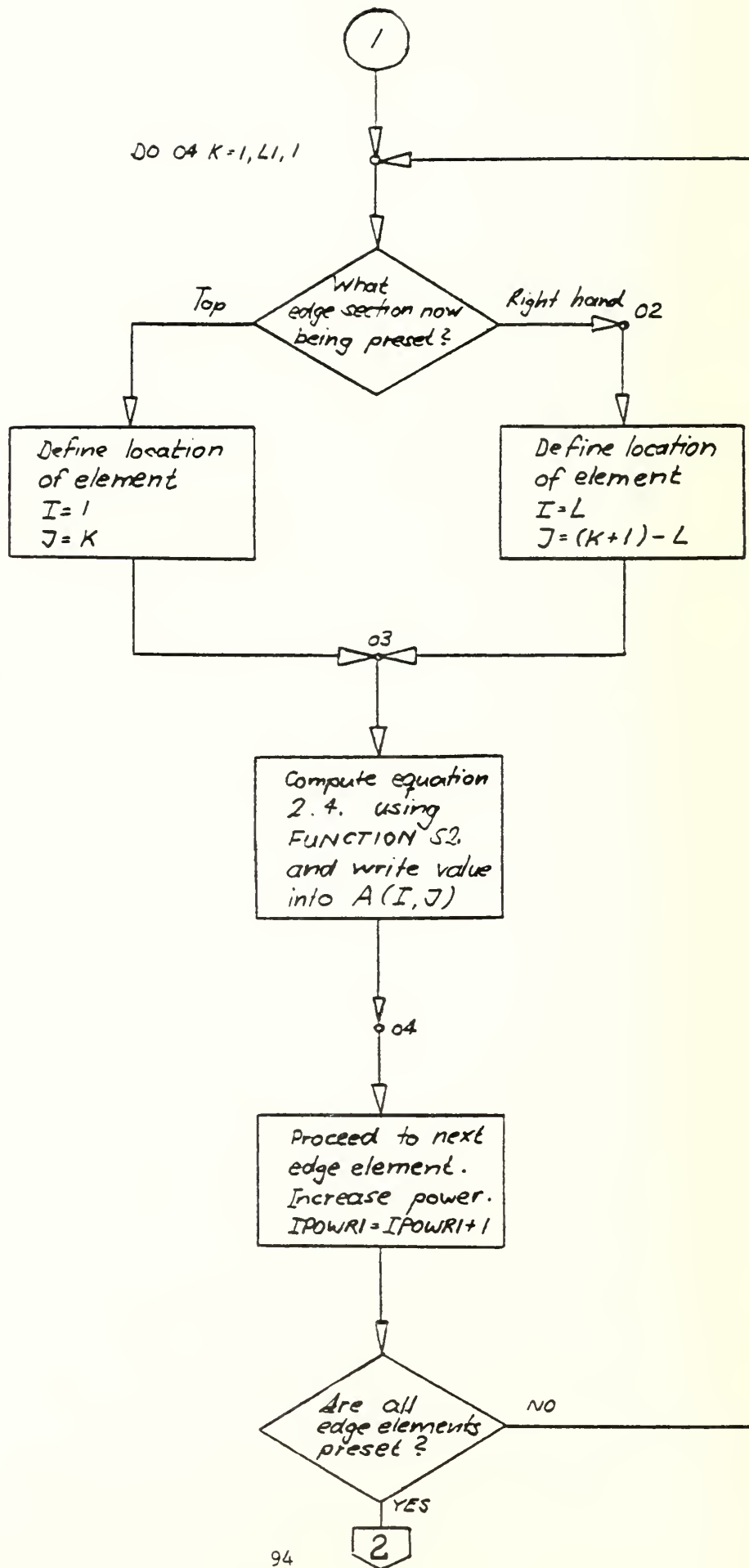
APPENDIX B: Software Description: Flow Charts

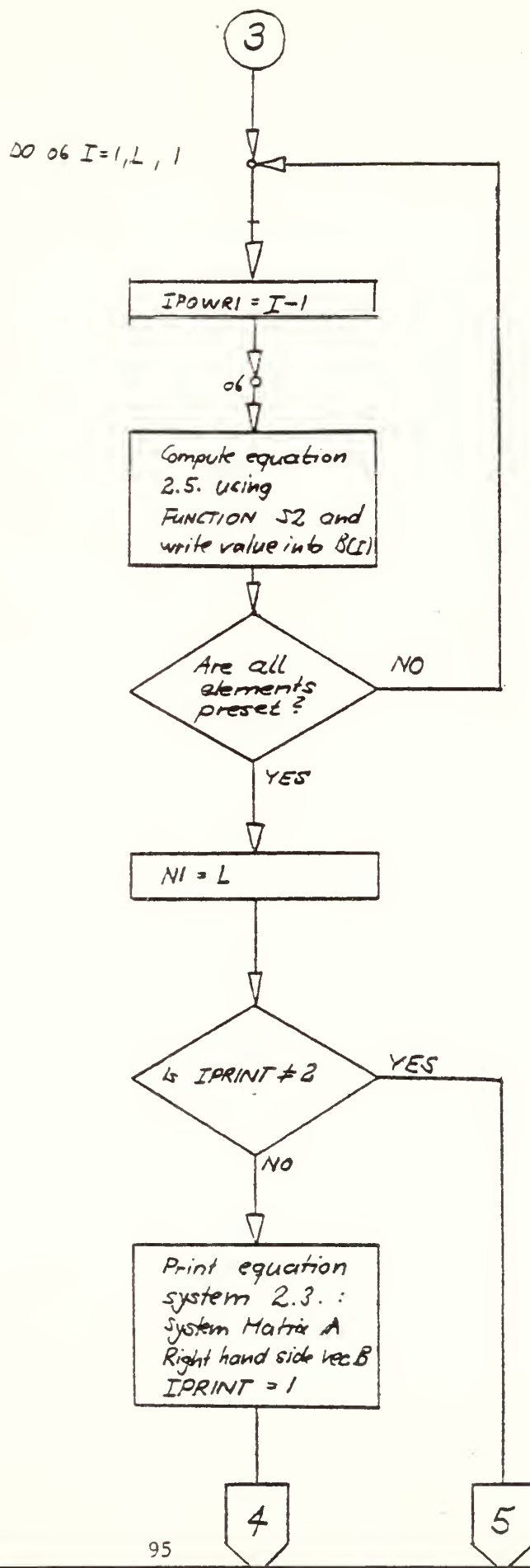
The following are given:

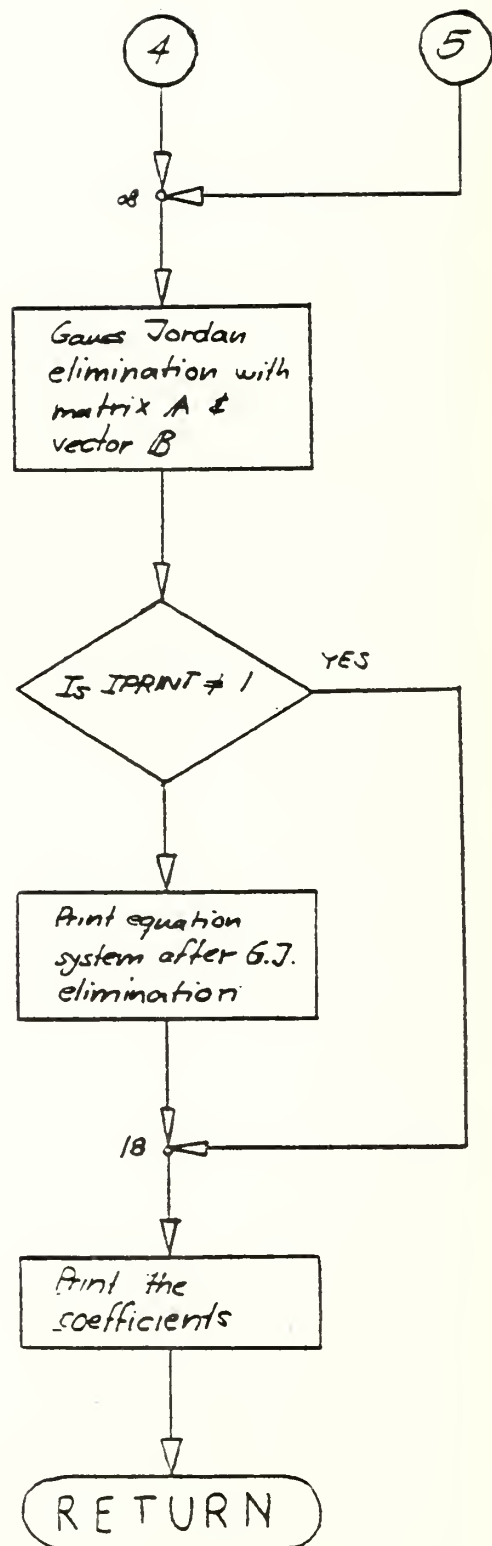
<u>Flow Chart</u>	<u>Page</u>
MAT2	93
S2	97
MAT3	98
MAT31	106
S3	109
MAT4	111
MAT41	119
MAT42	124
S4	127

Flow chart MAT2

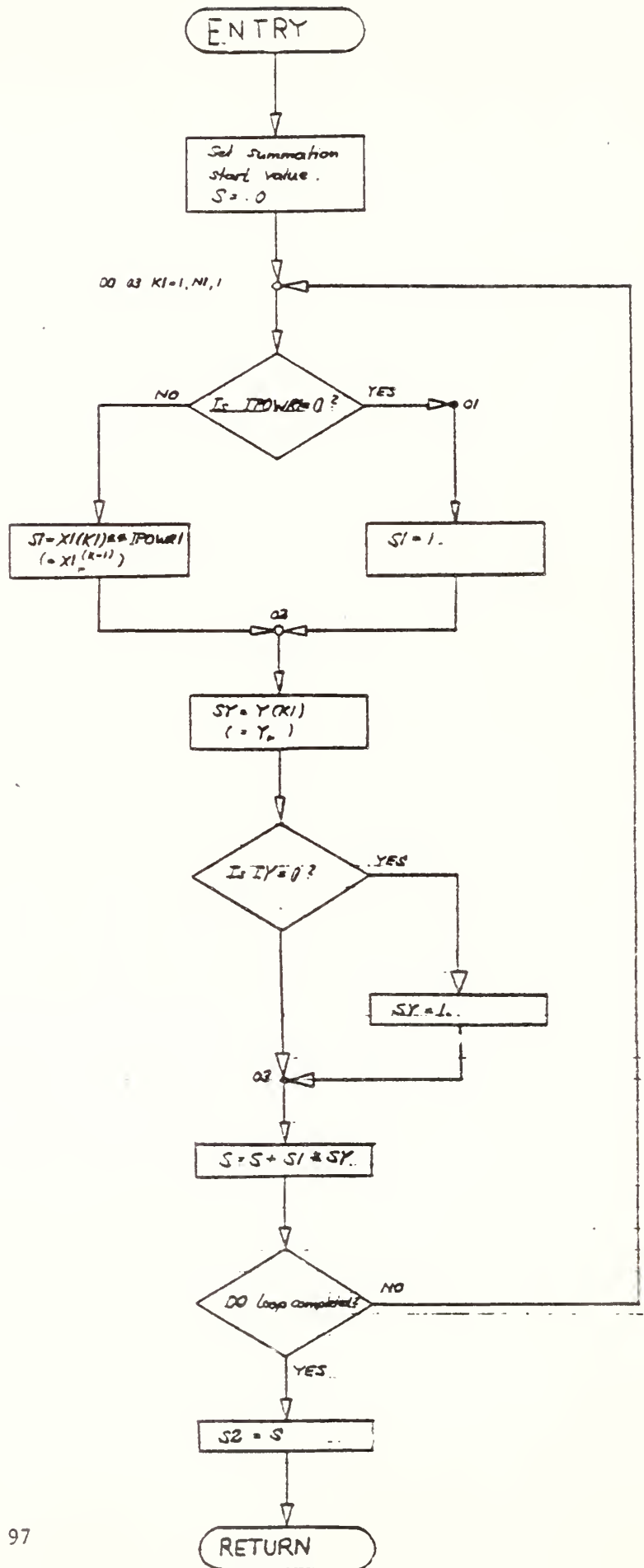




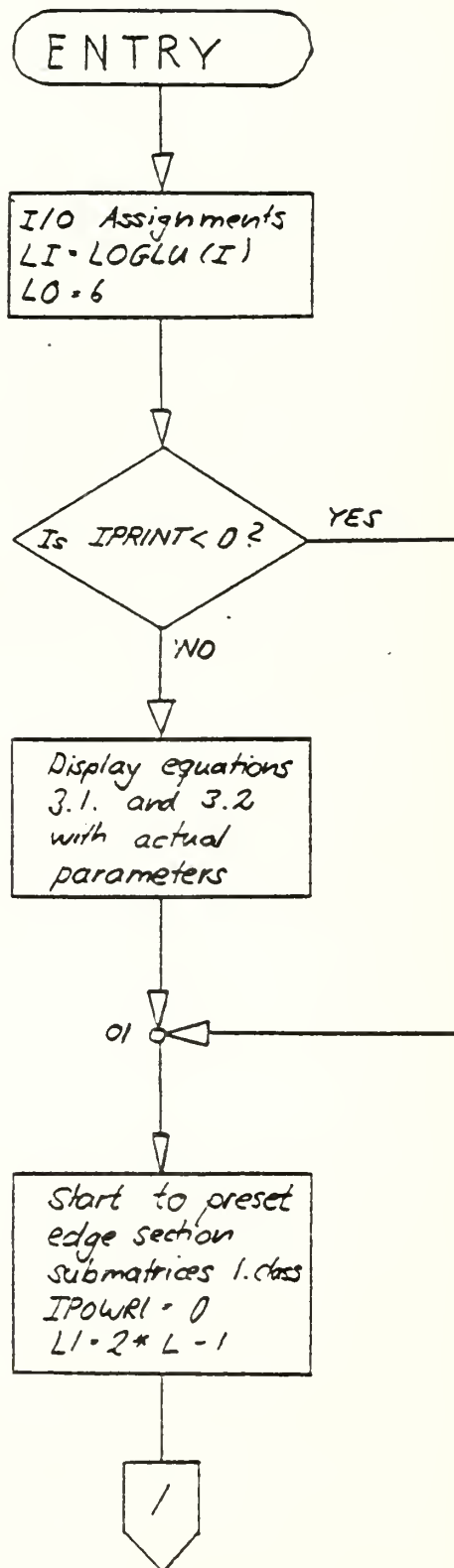


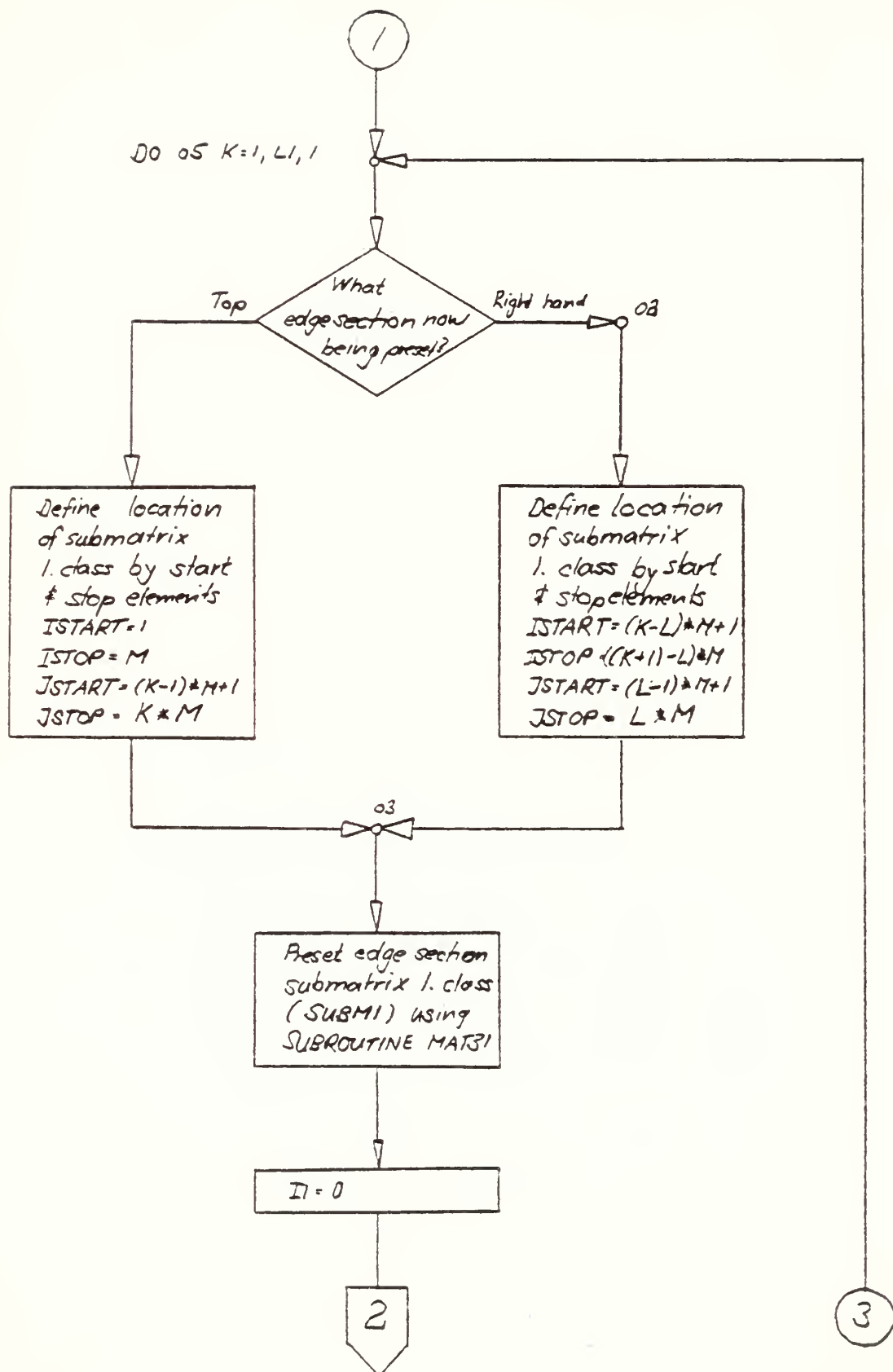


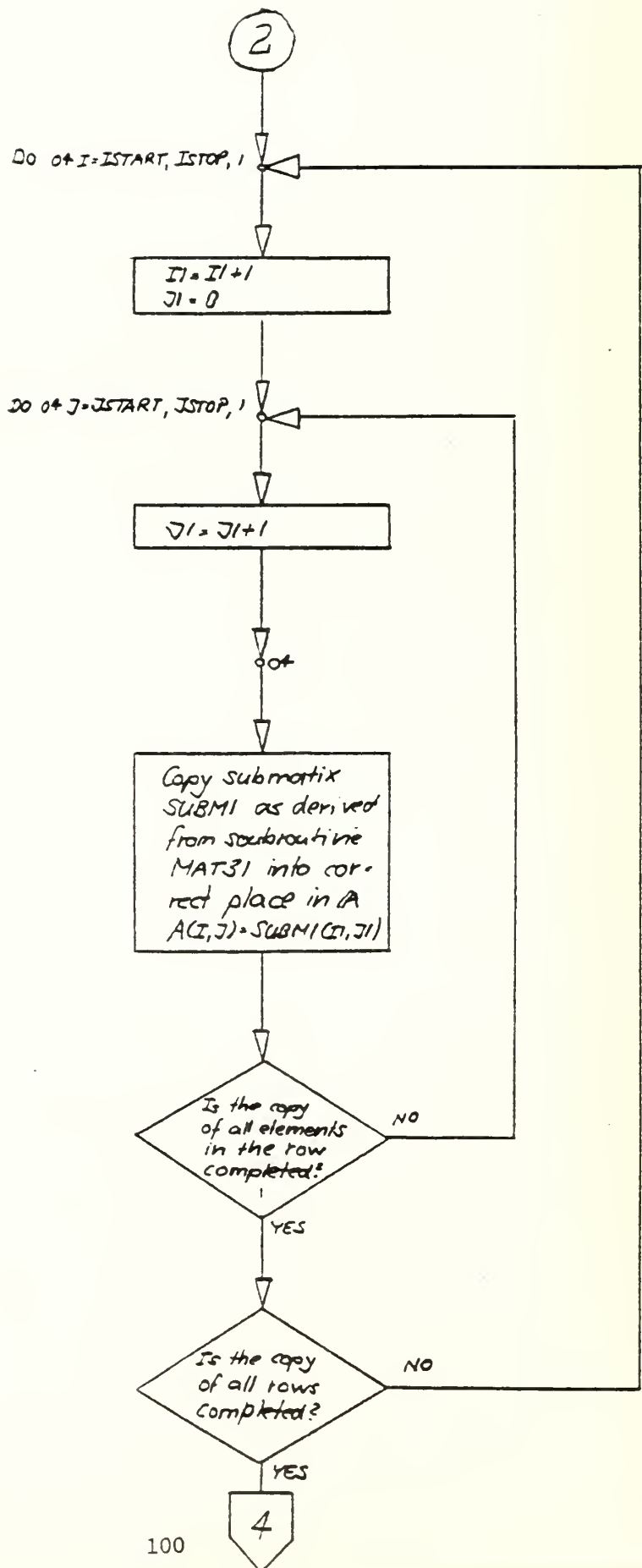
Flow chart S2

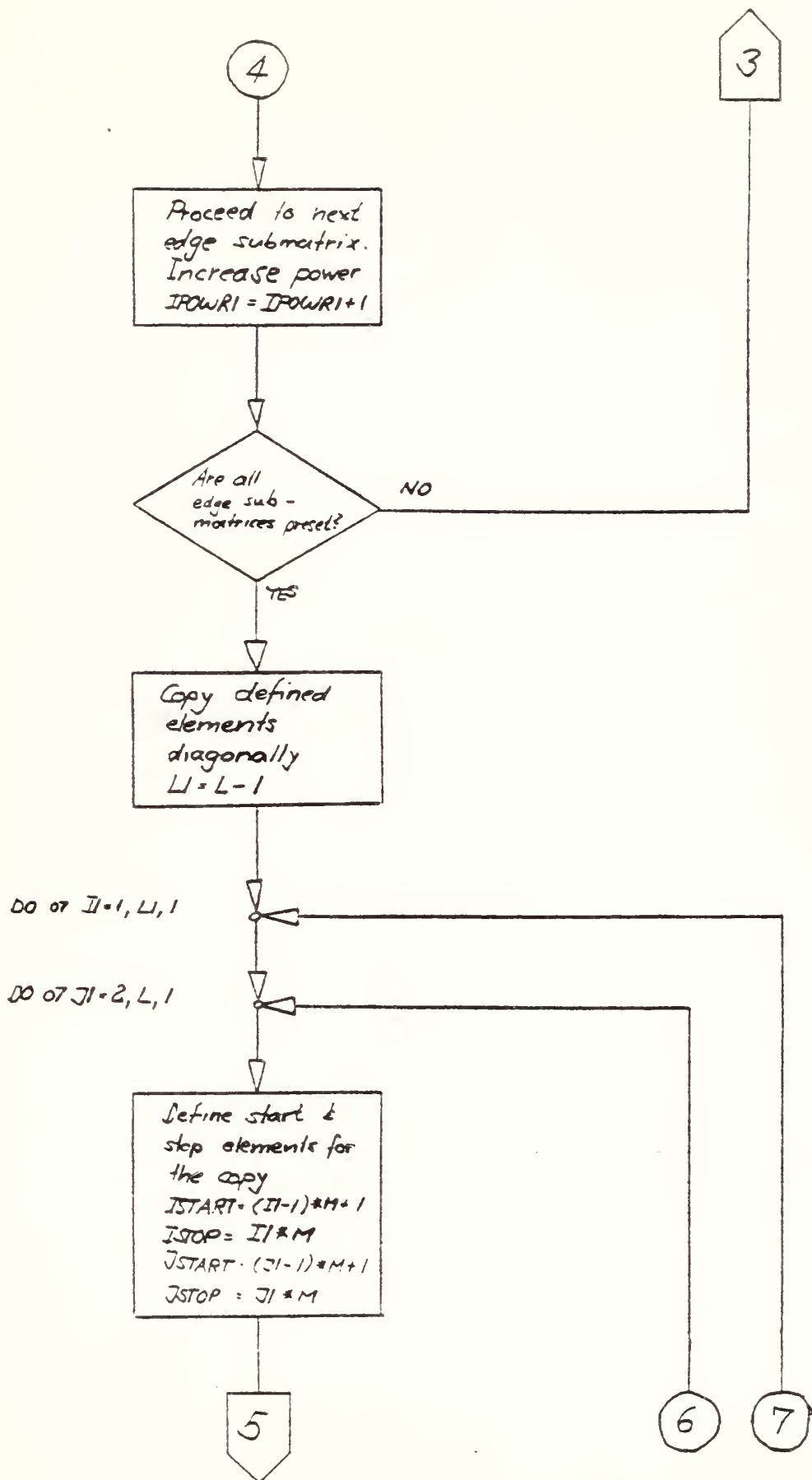


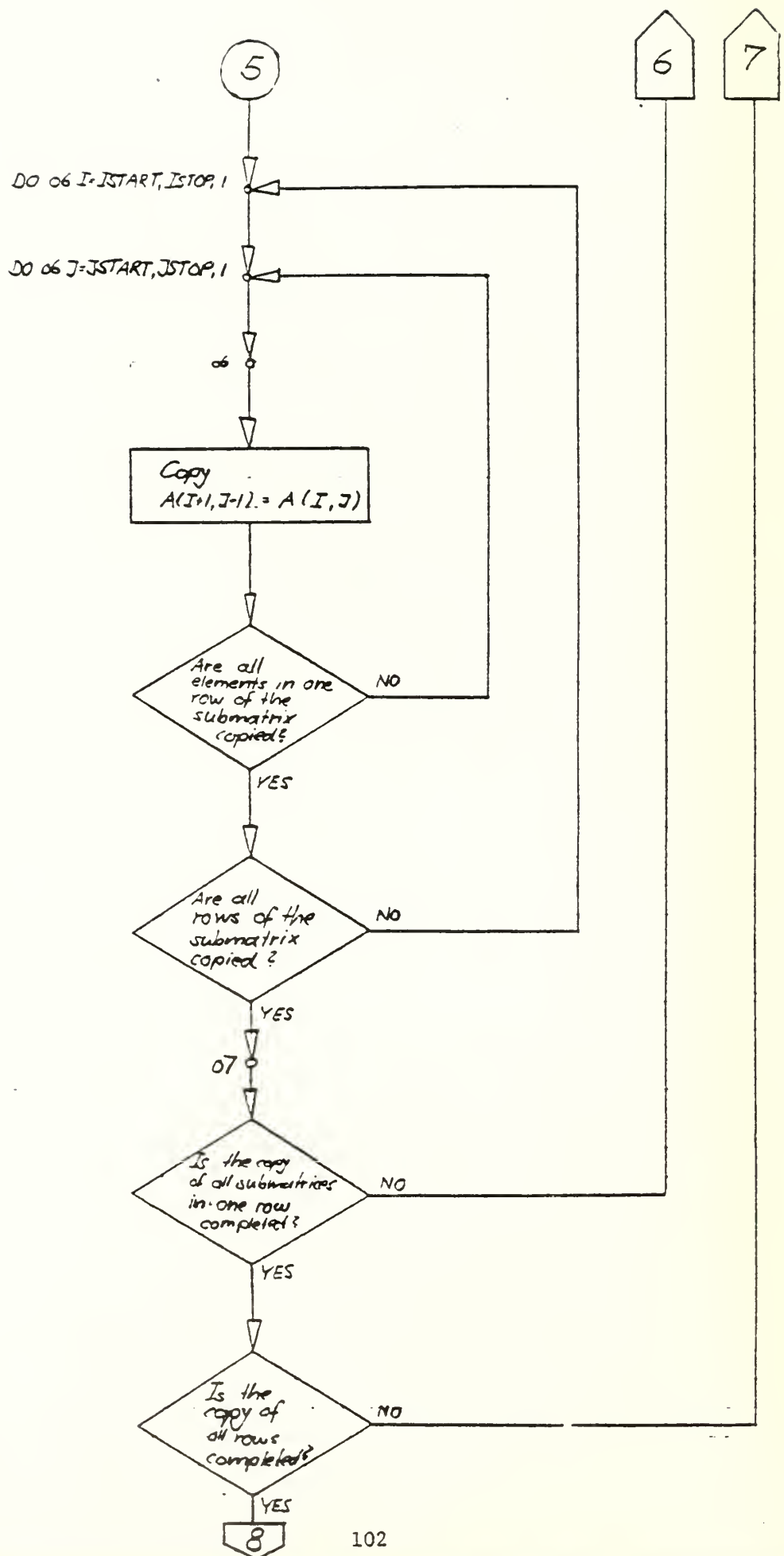
Flow chart MAT3

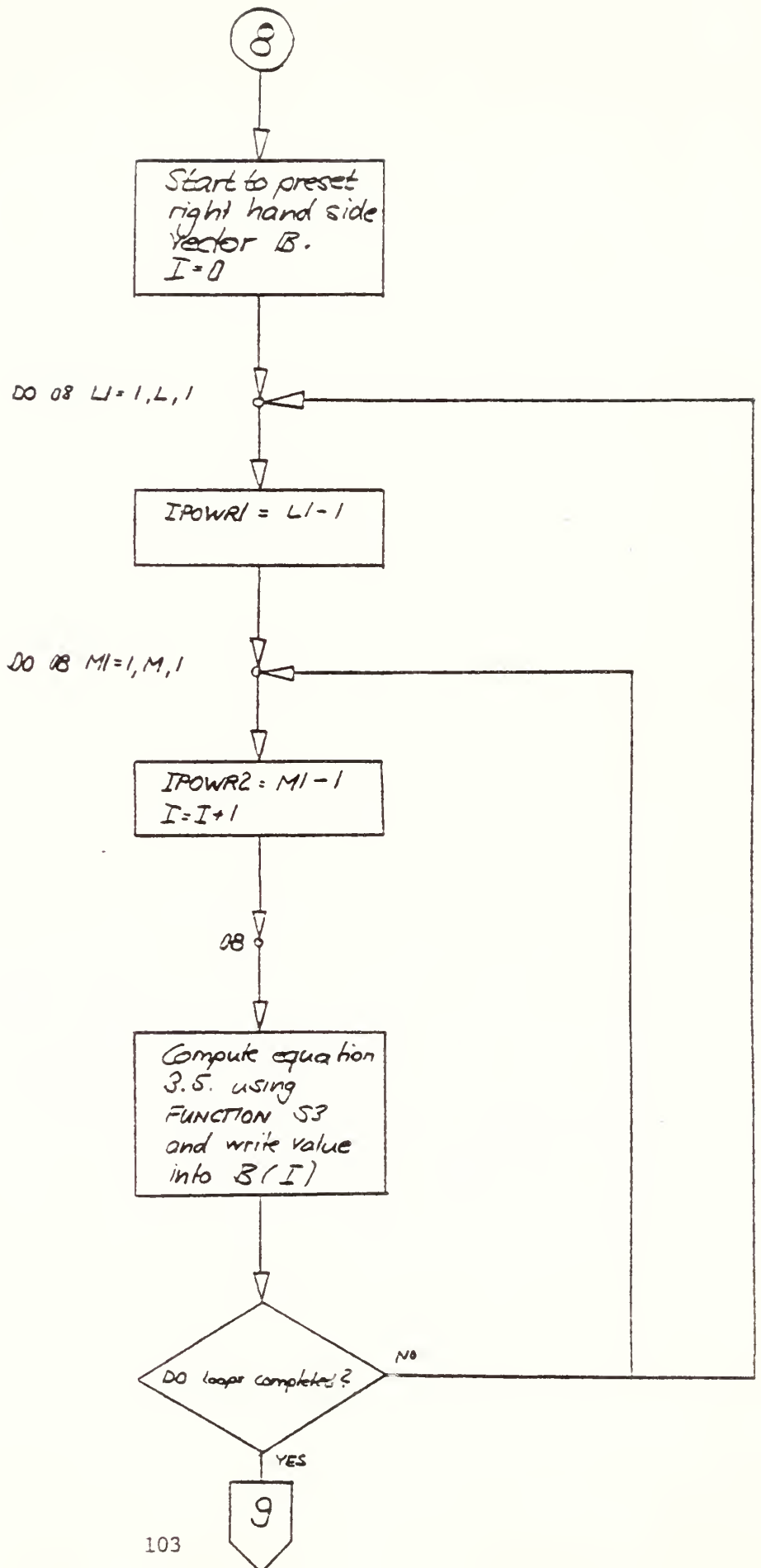


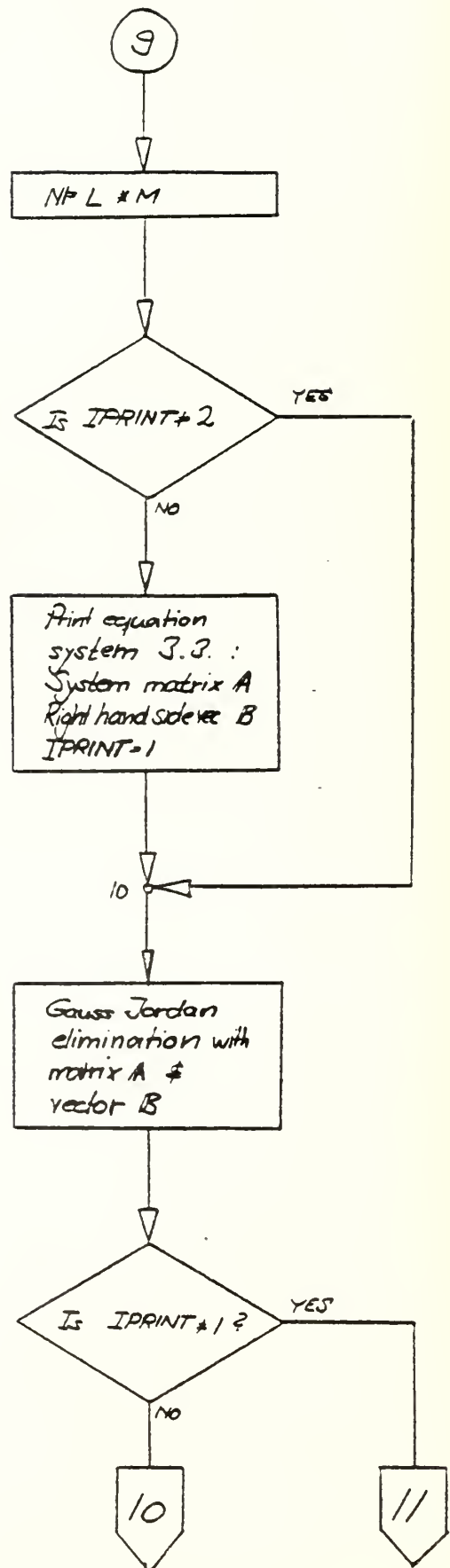


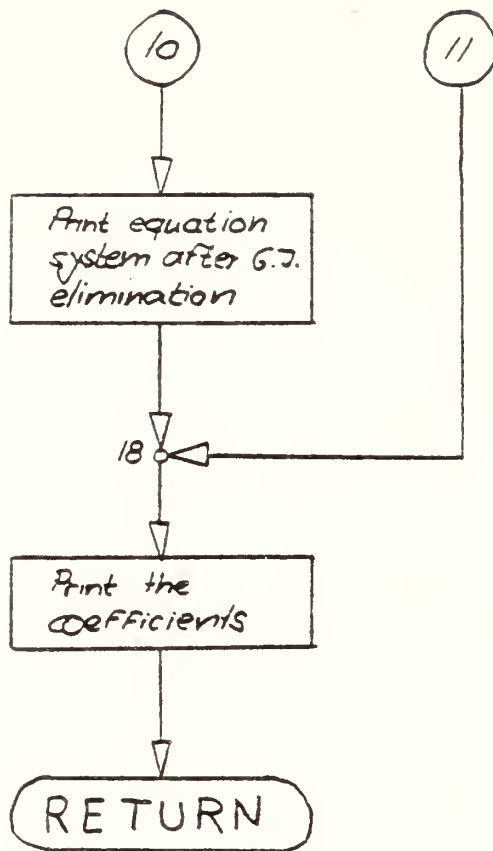




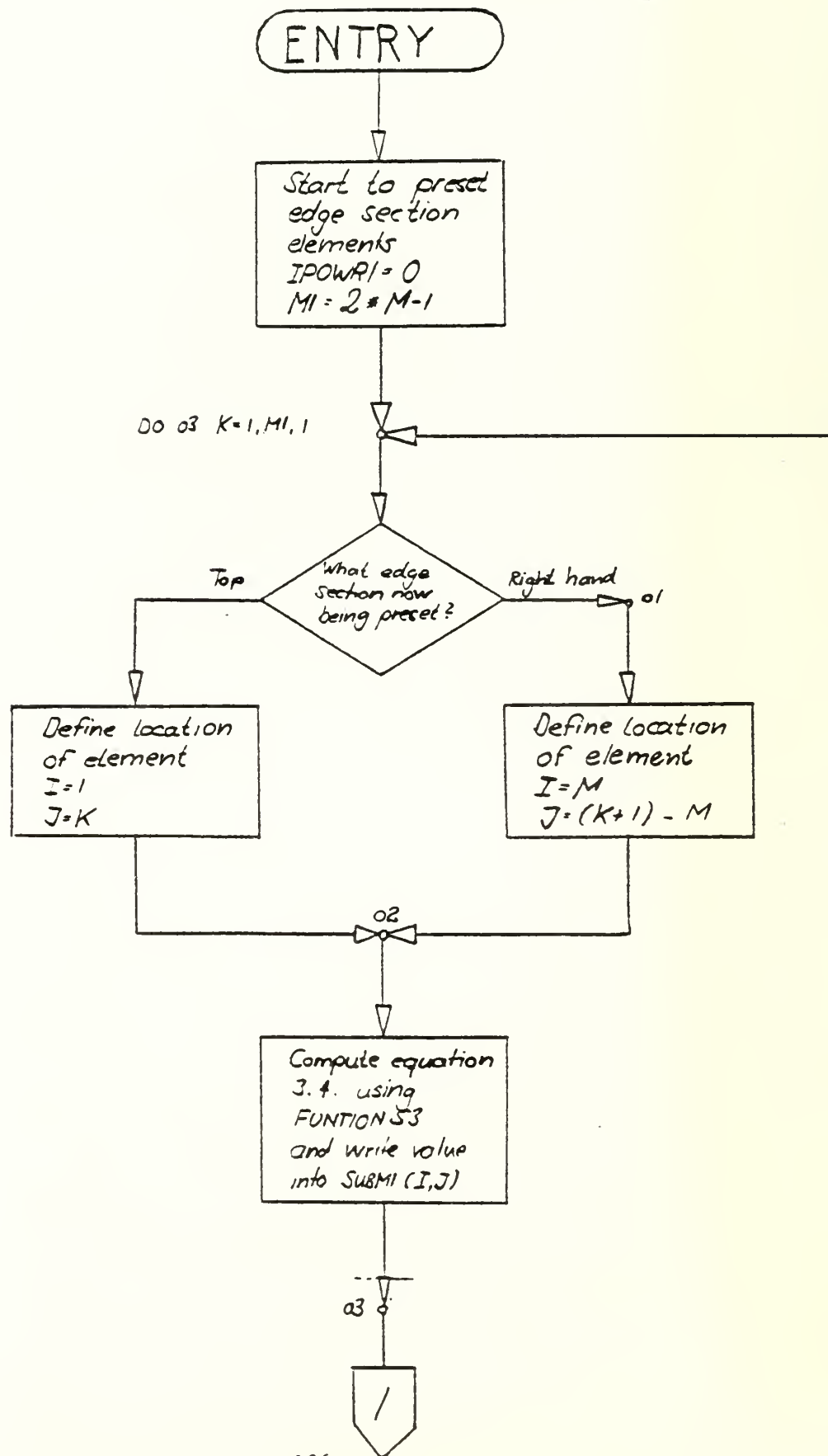


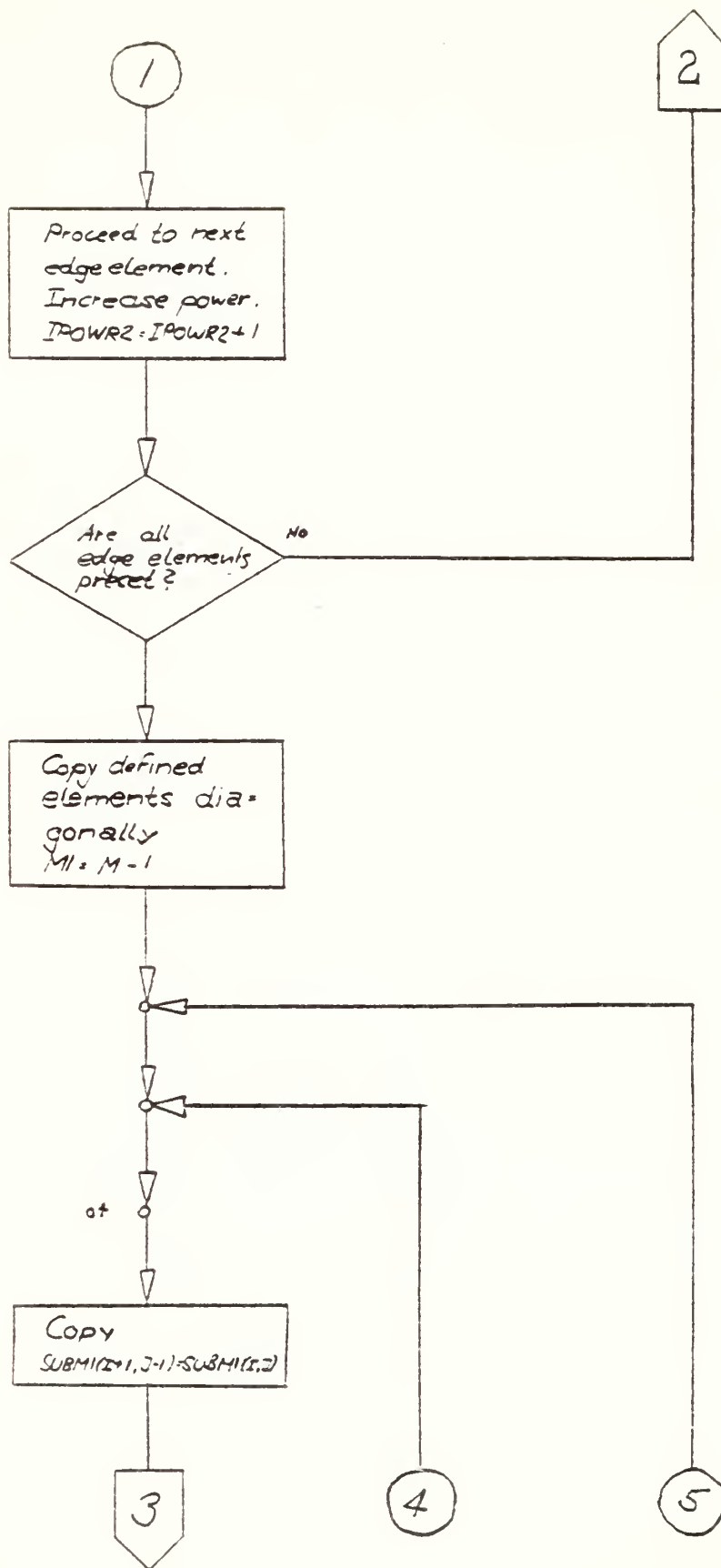


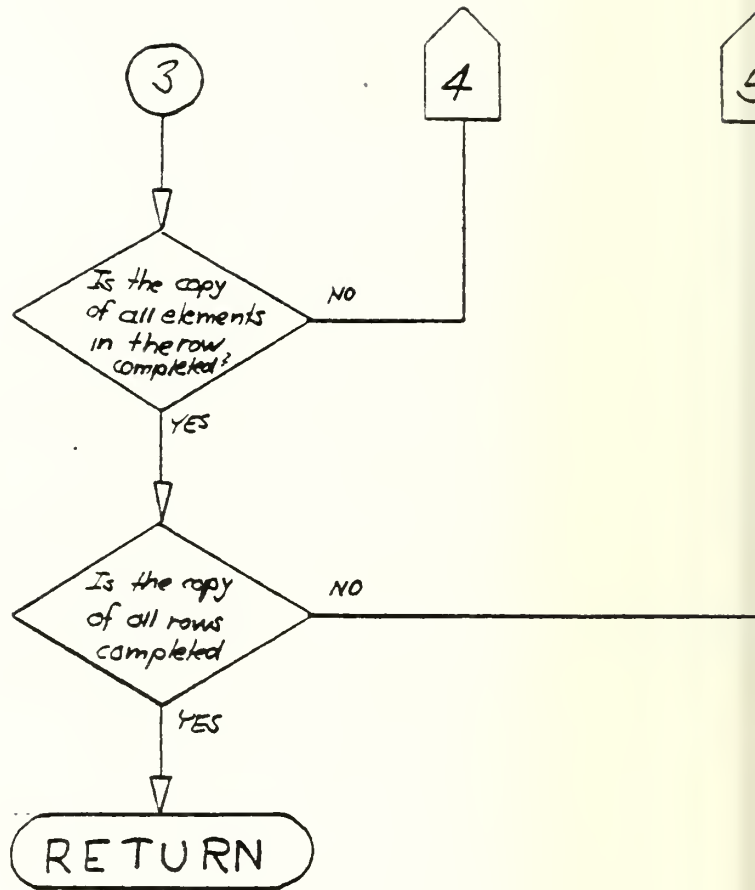




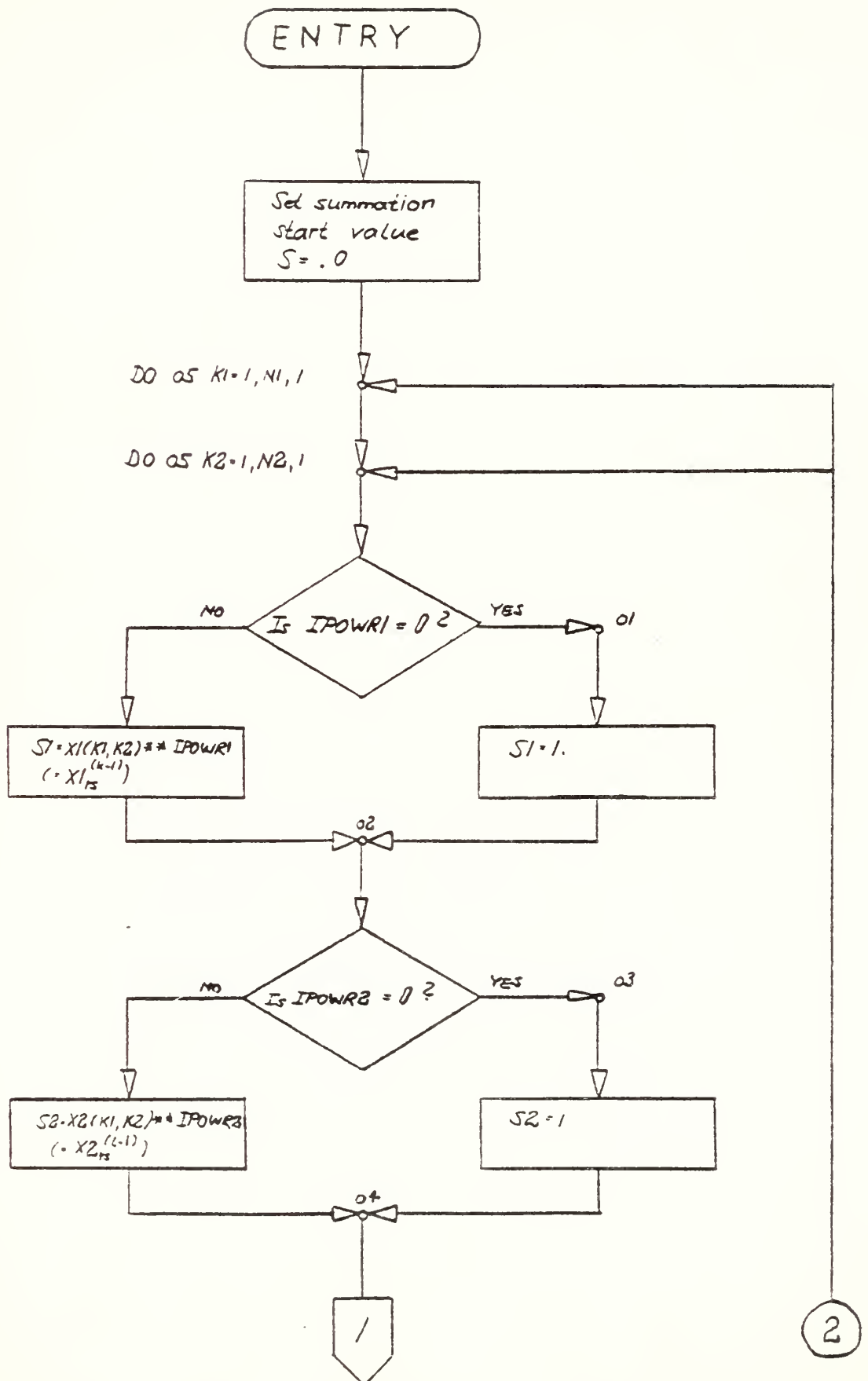
Flow chart MAT31

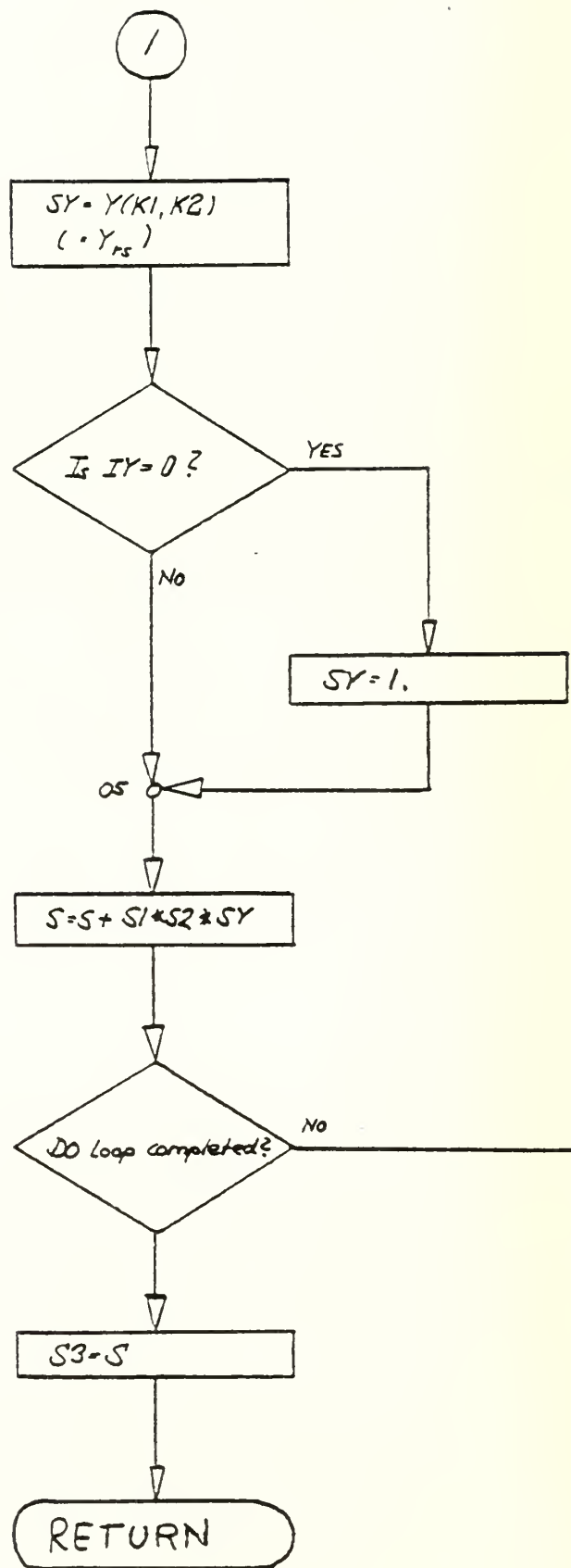




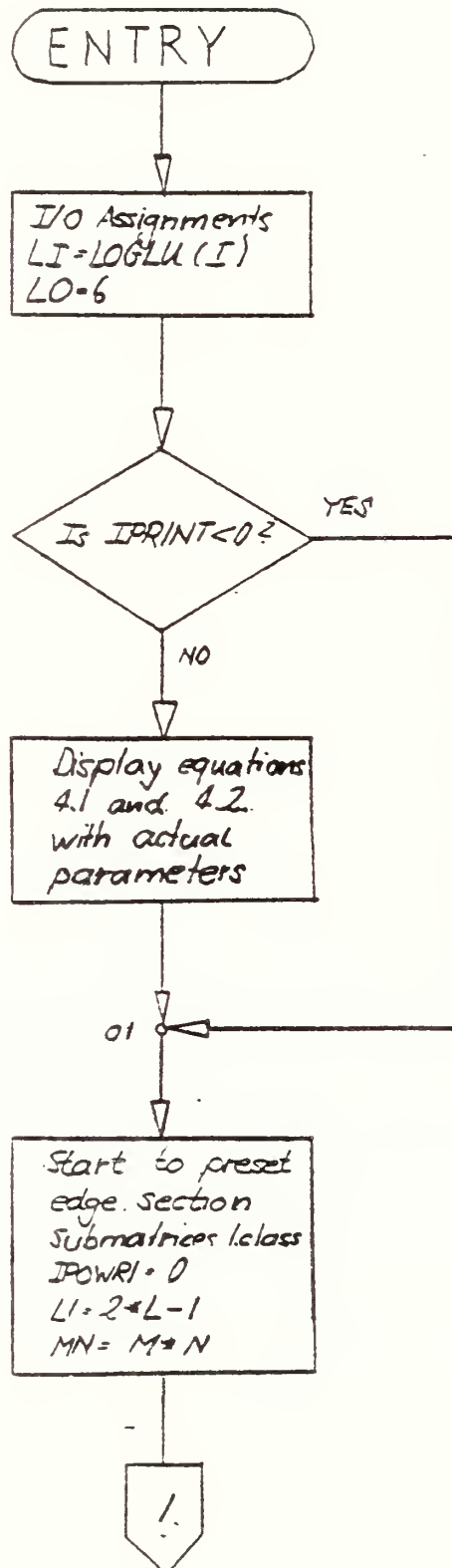


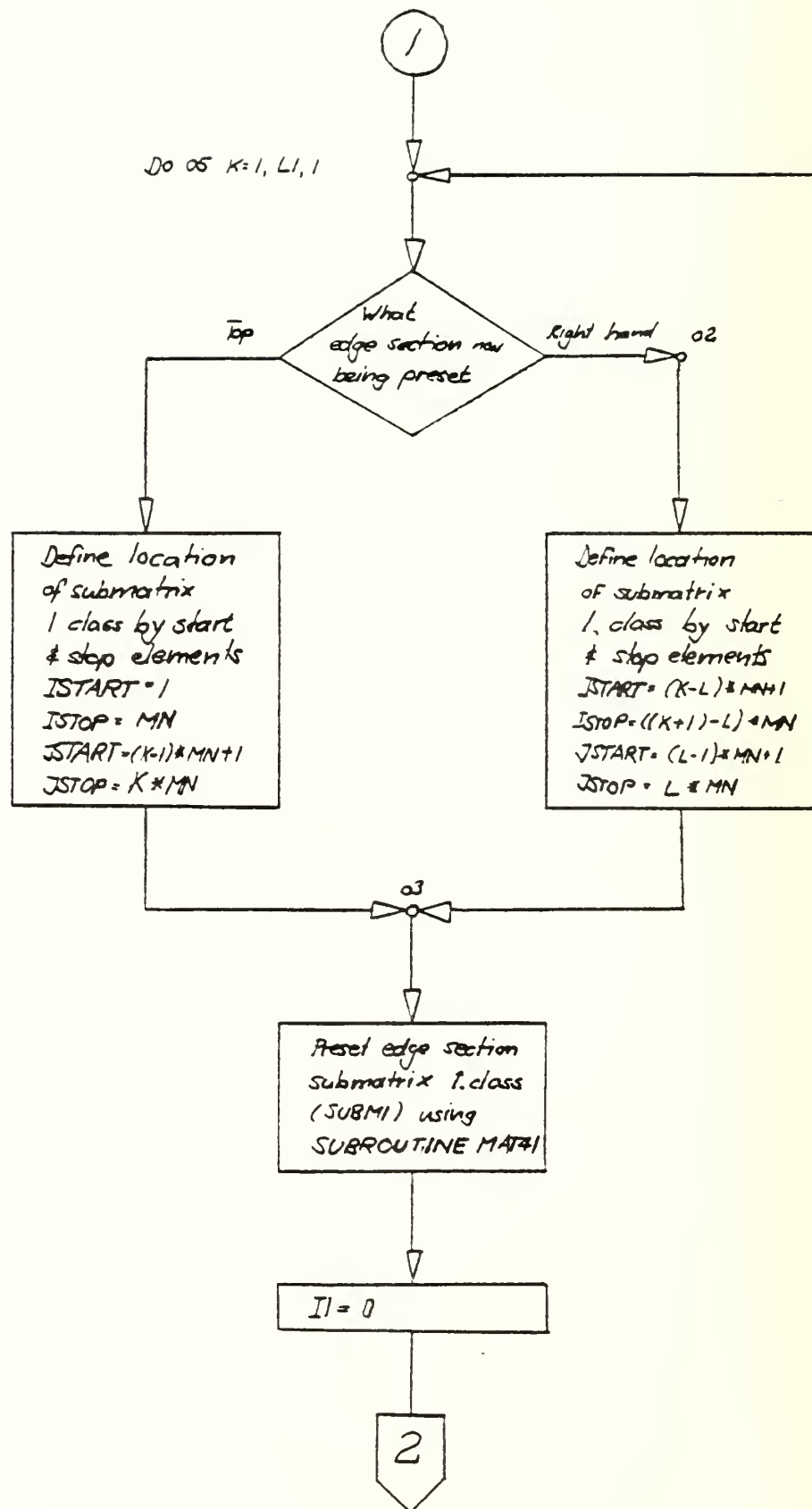
Flow chart S3

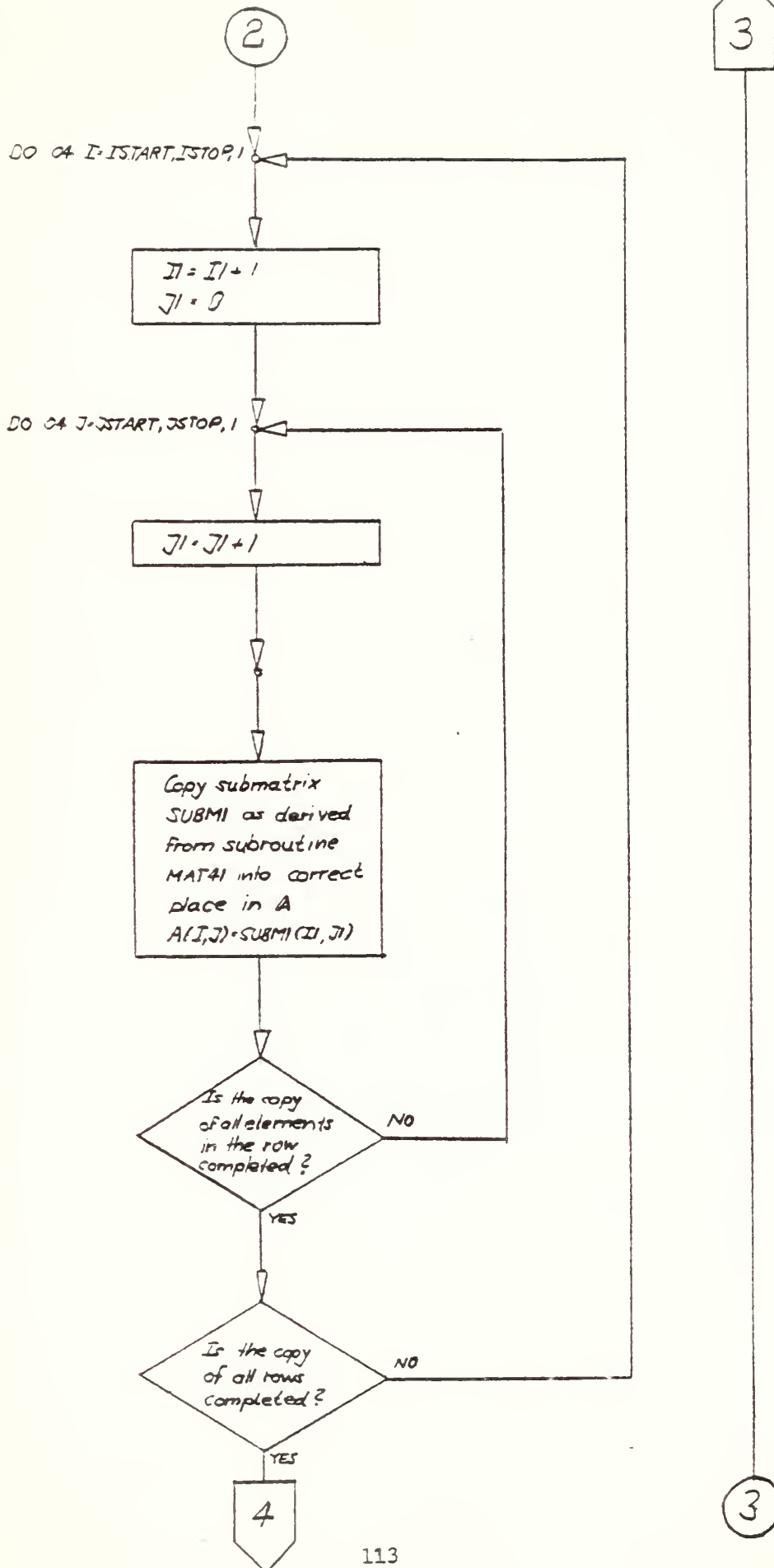


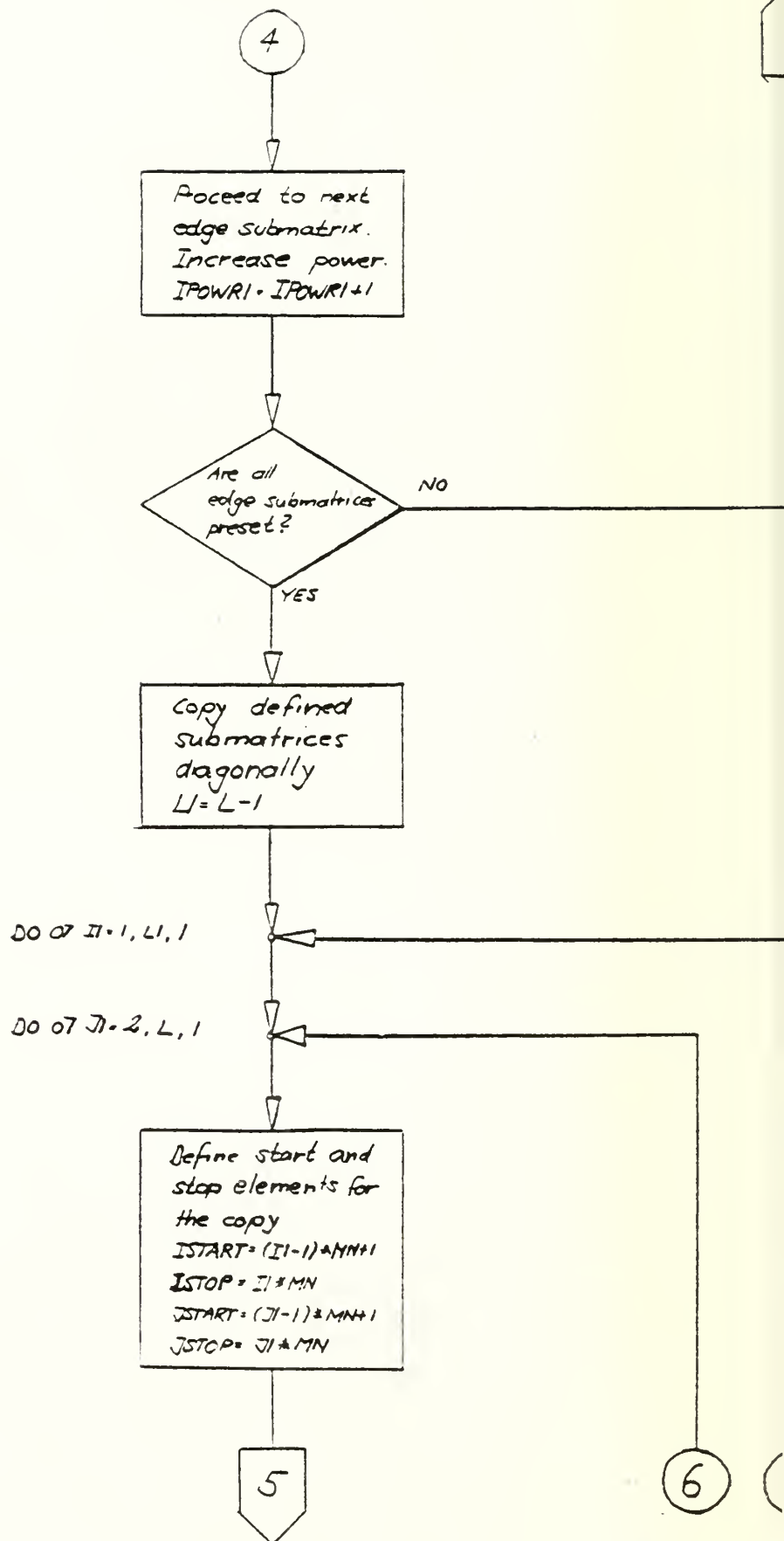


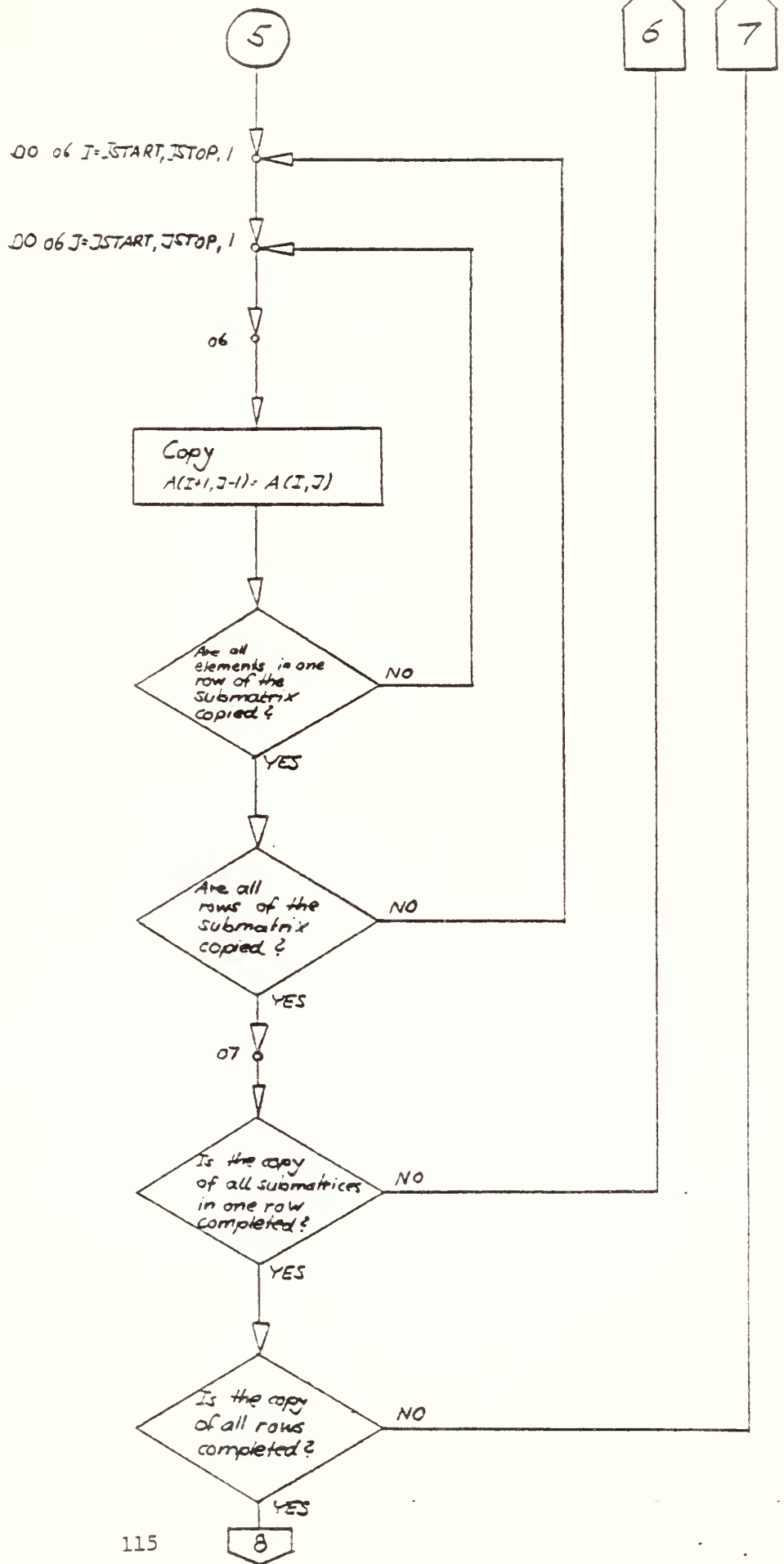
Flow chart MAT4

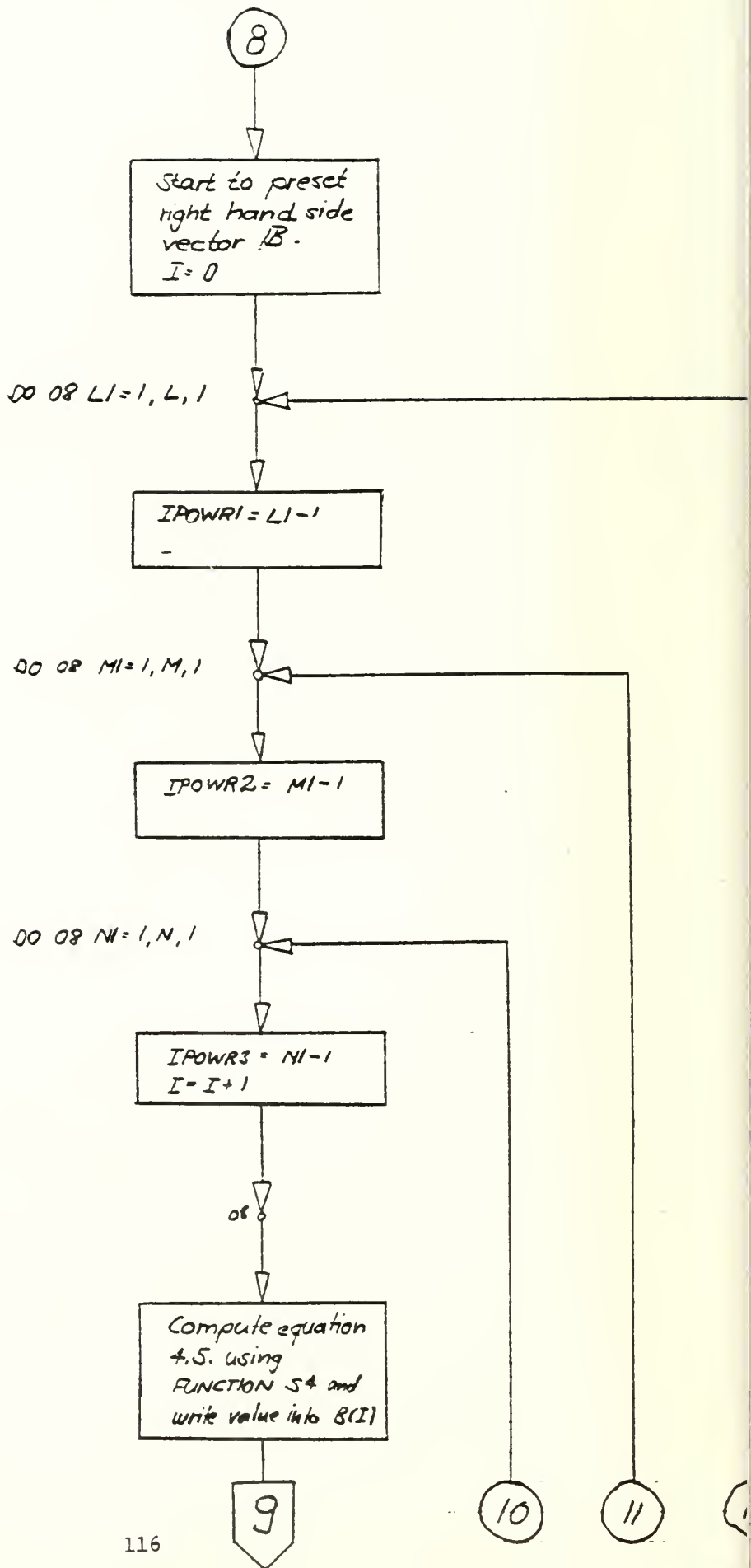


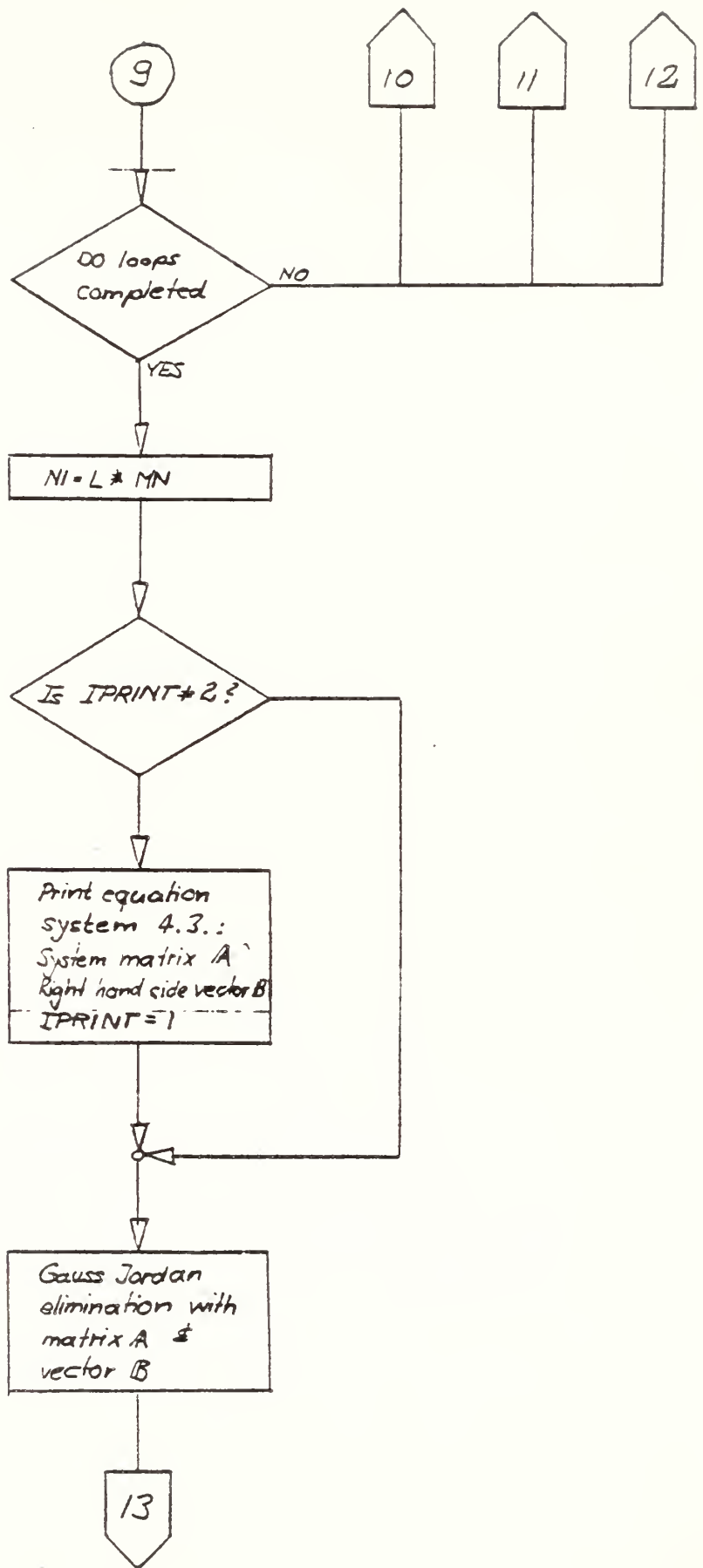


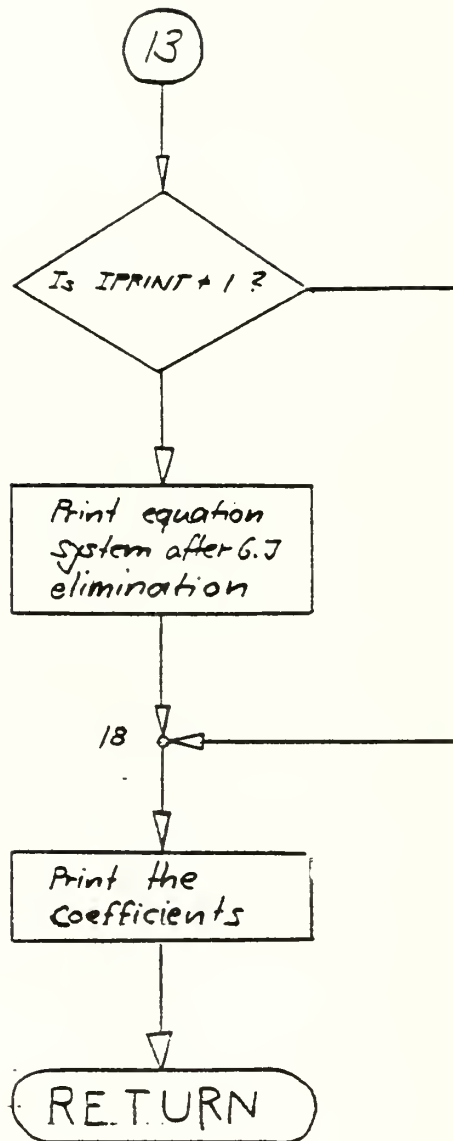




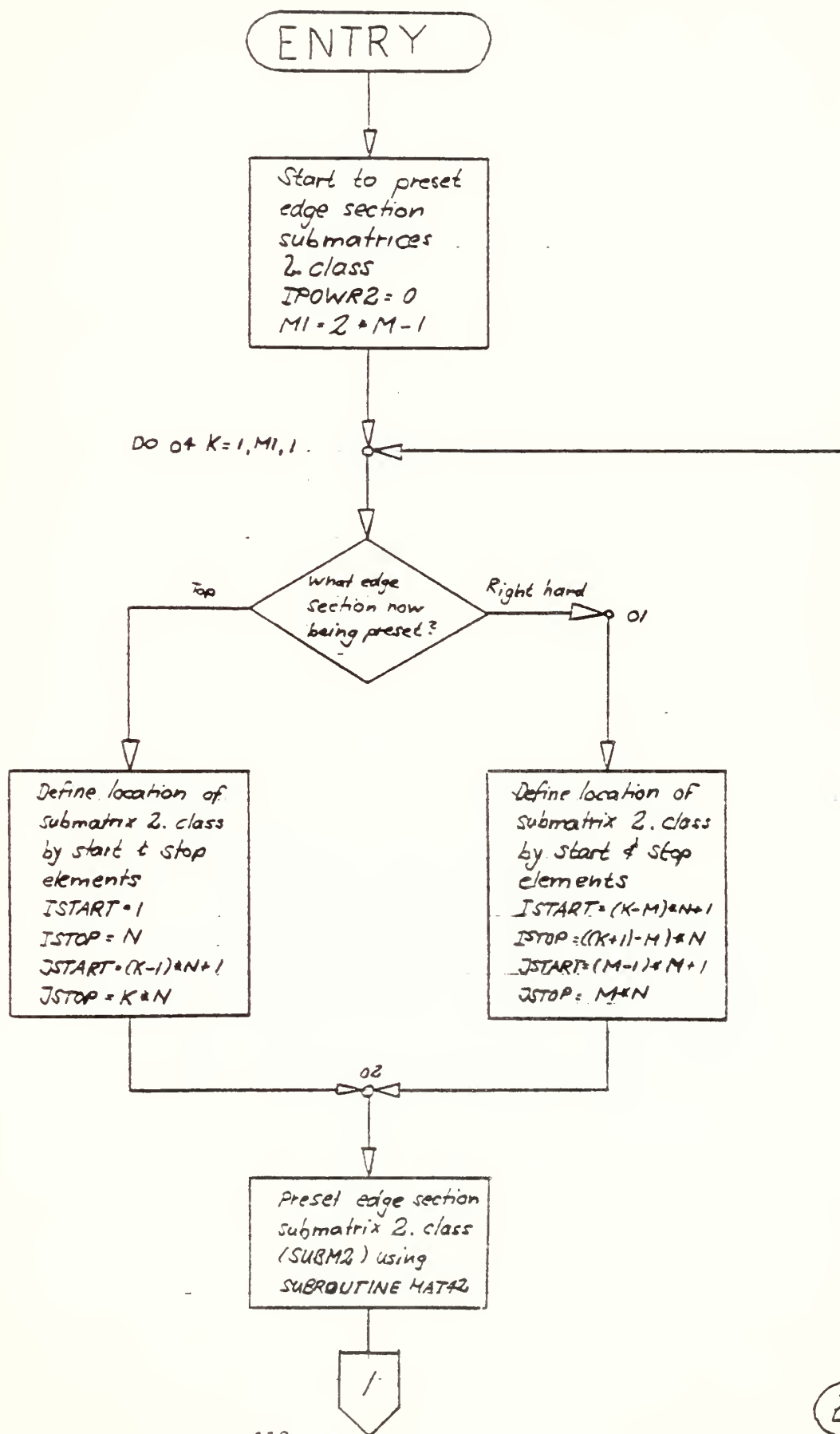


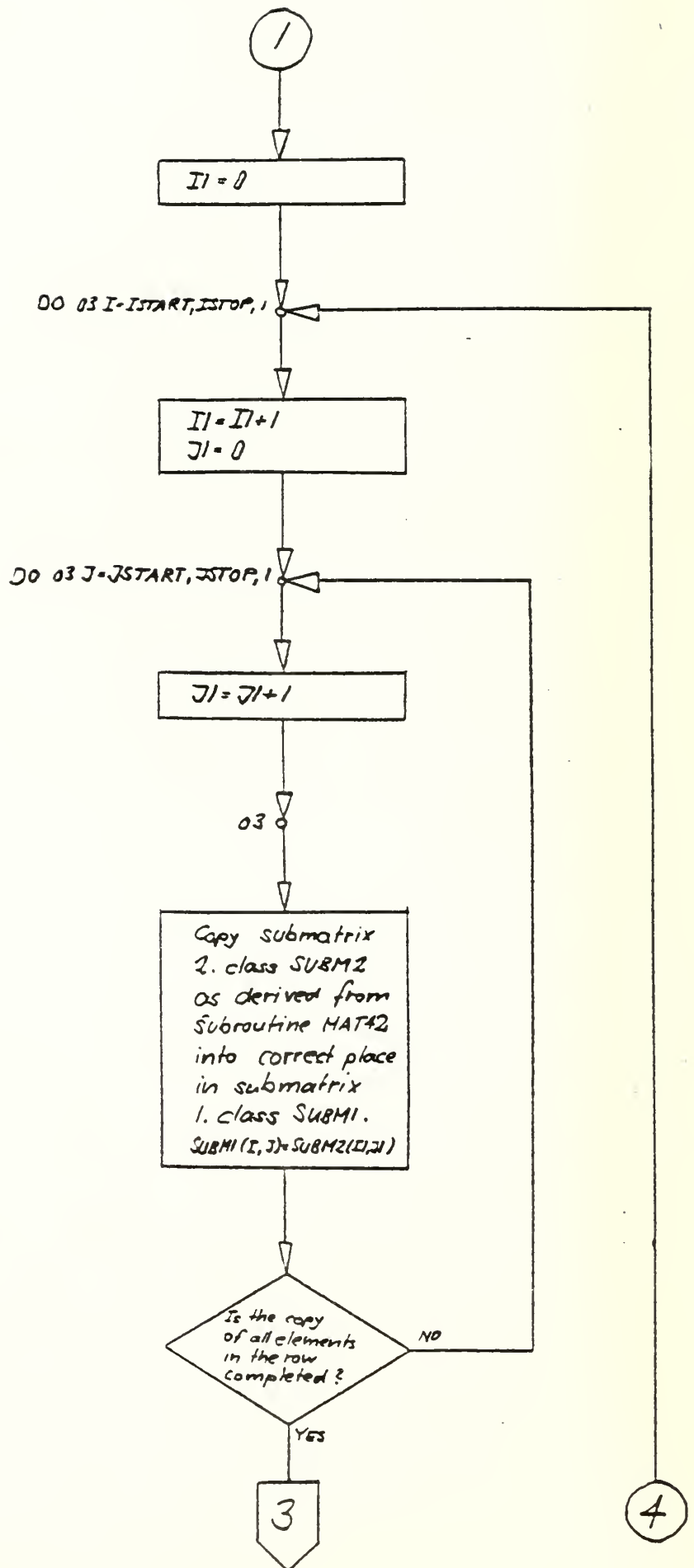


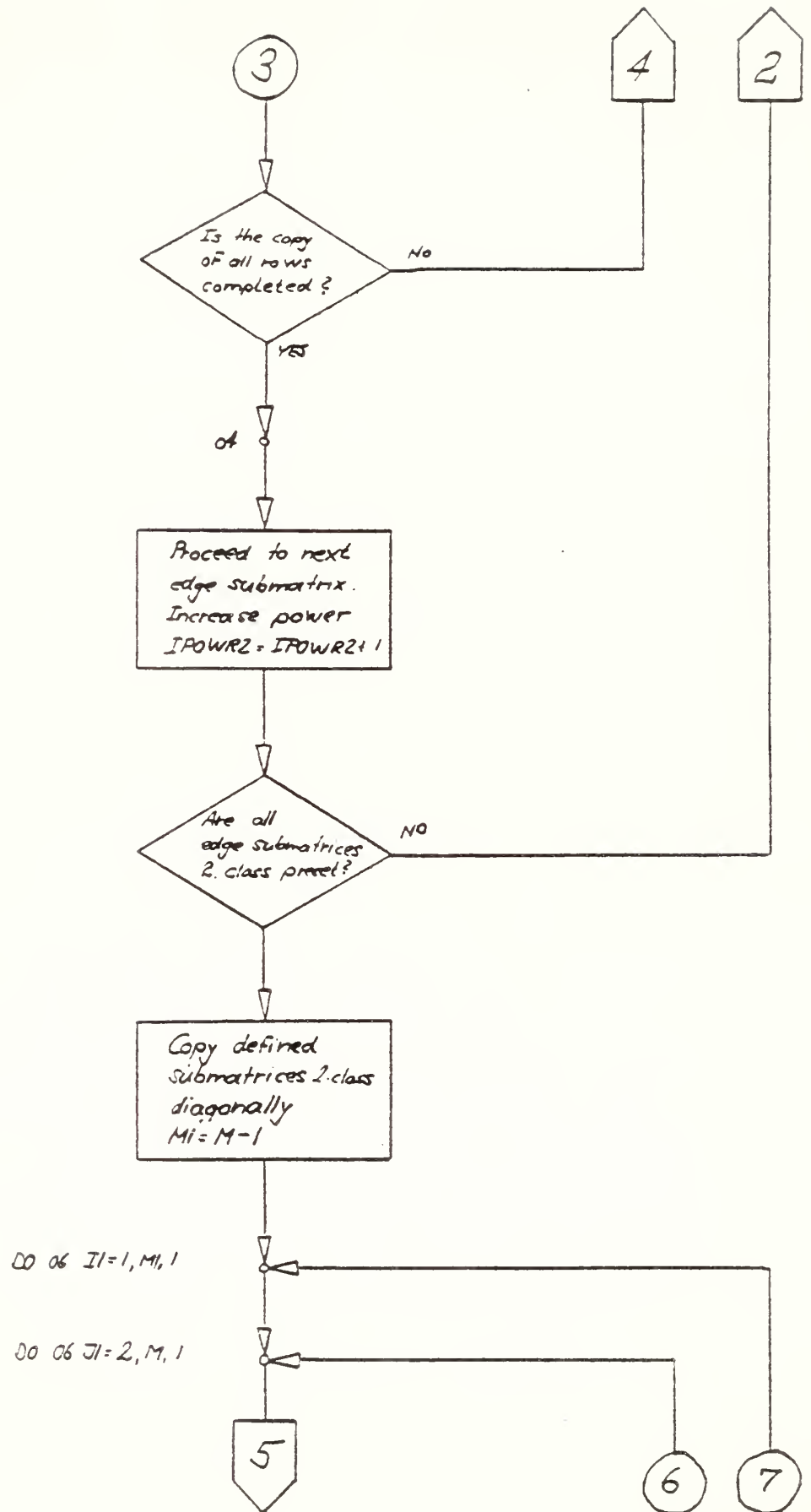


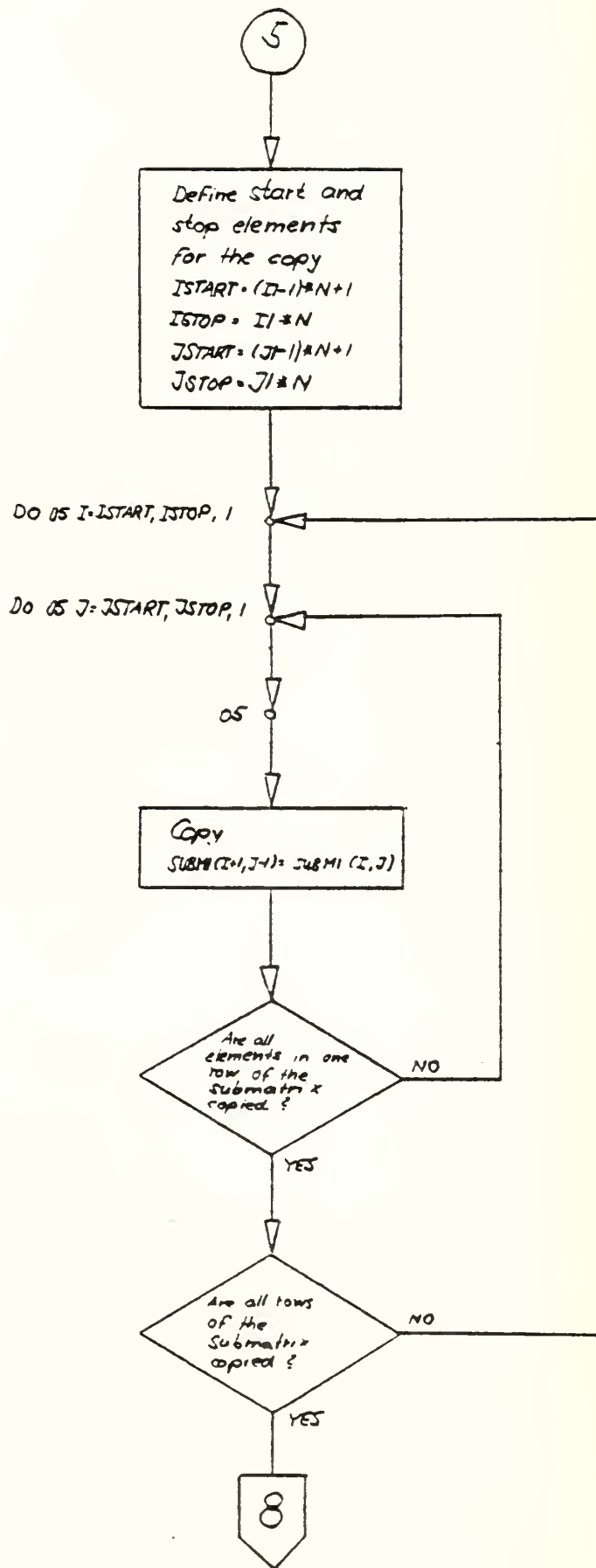


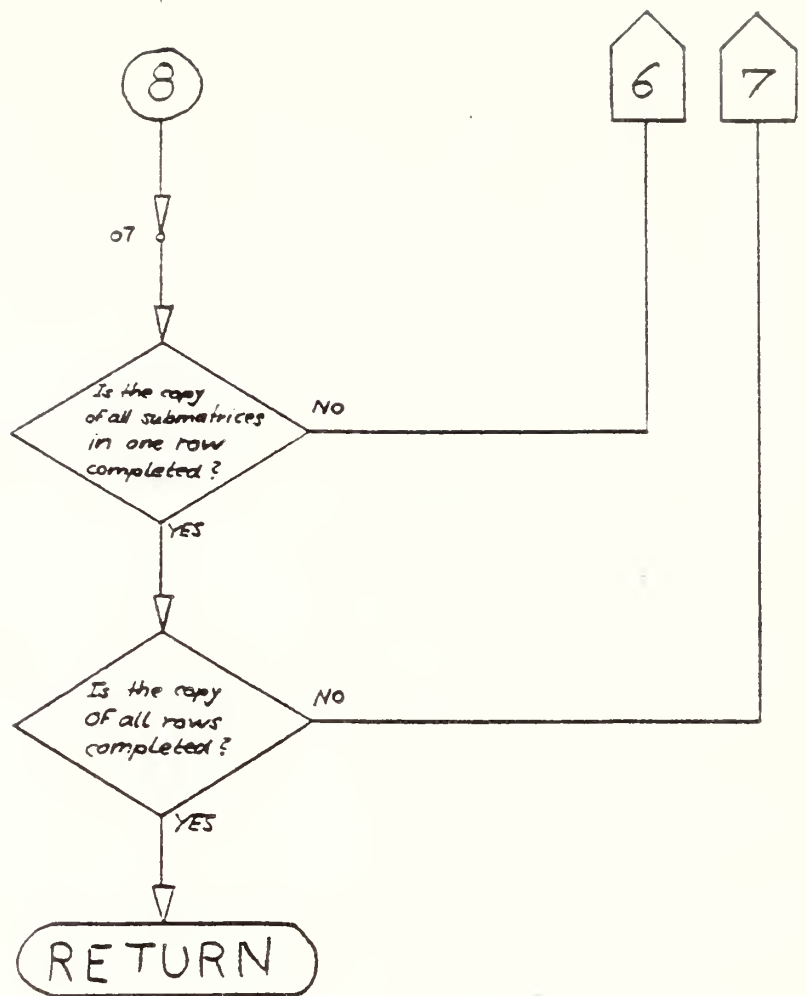
Flow chart MAT41



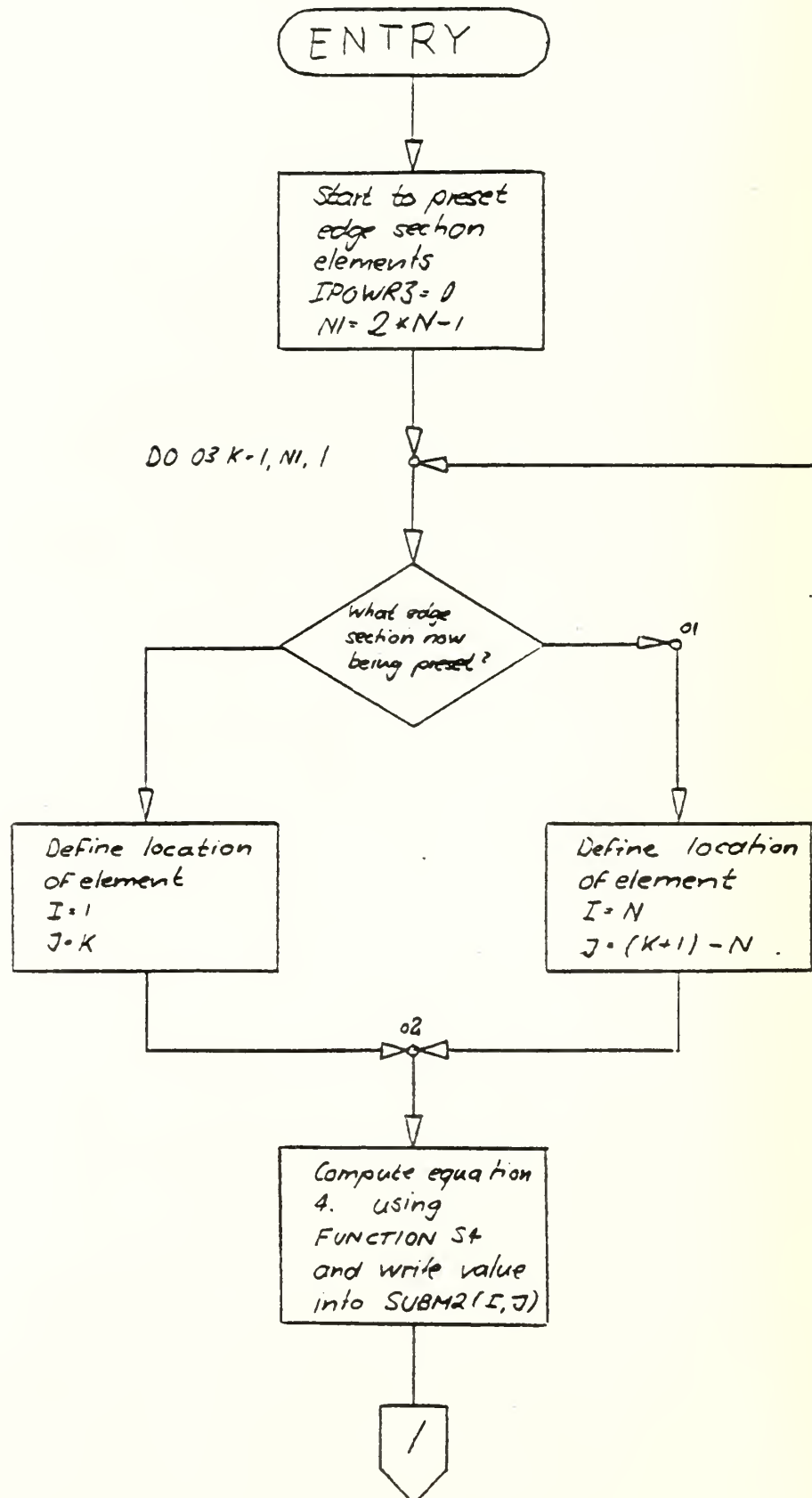


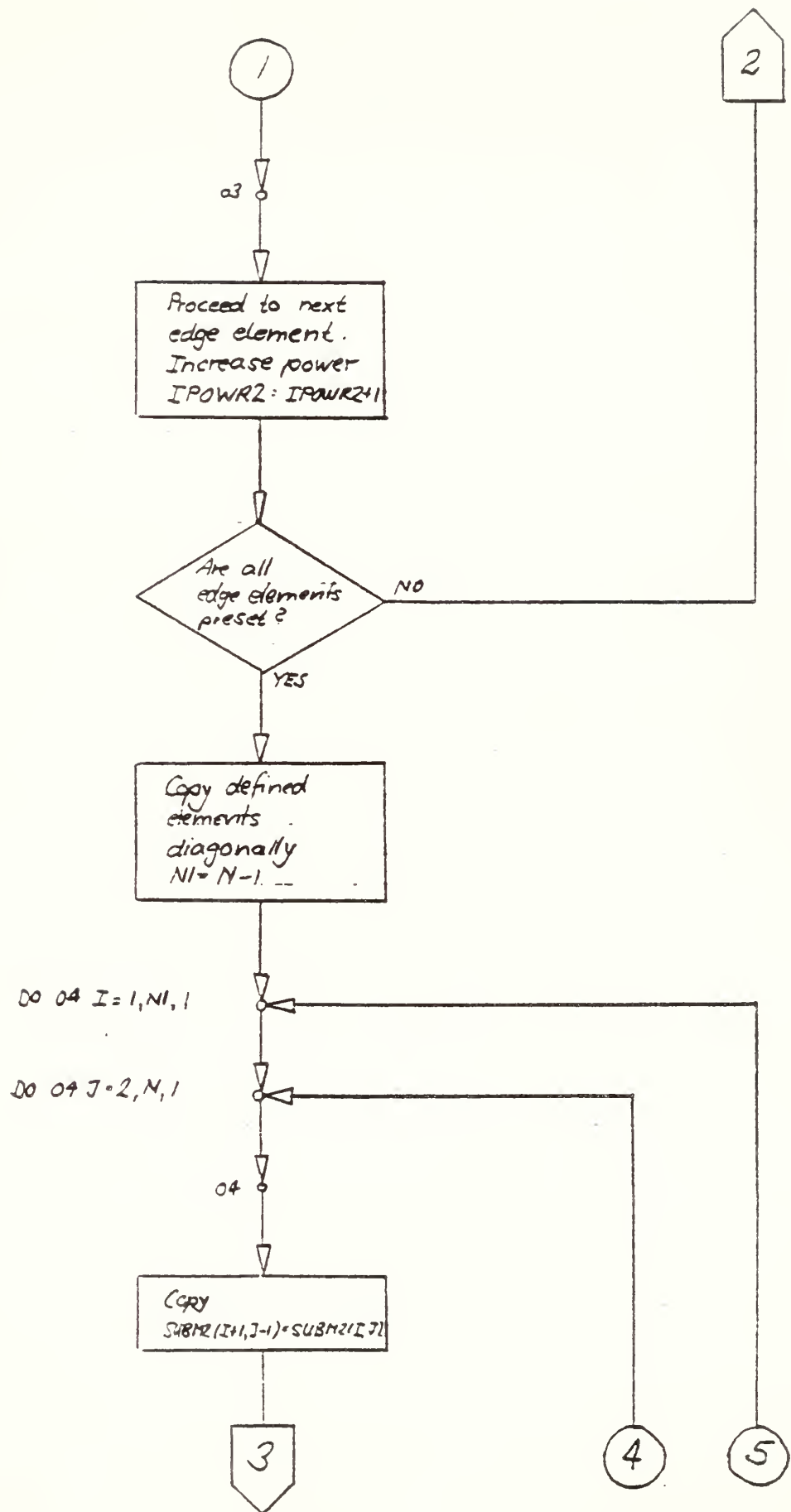


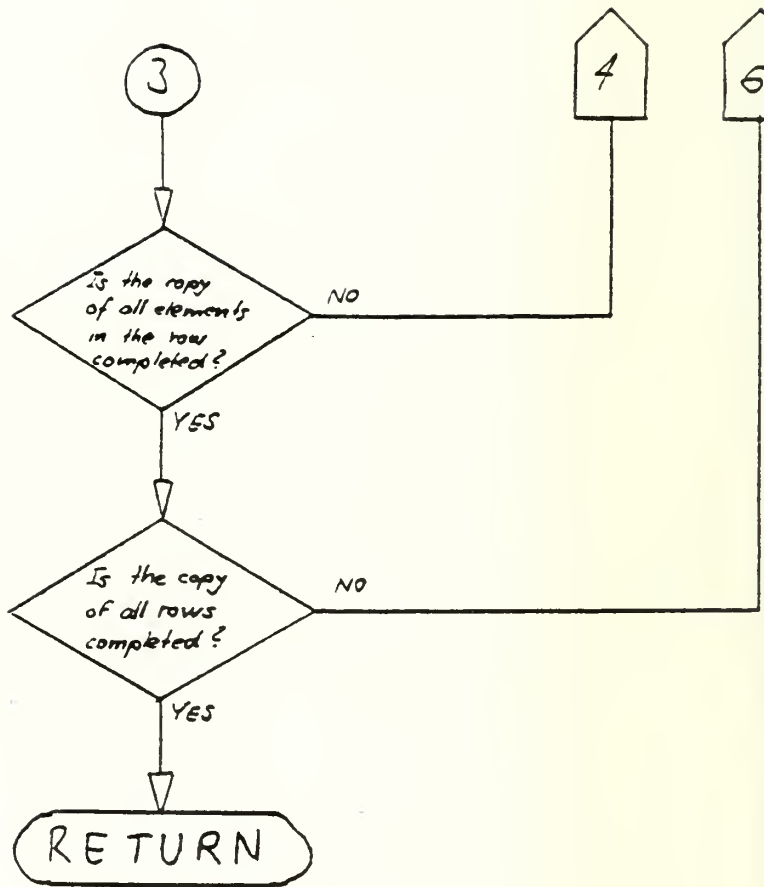




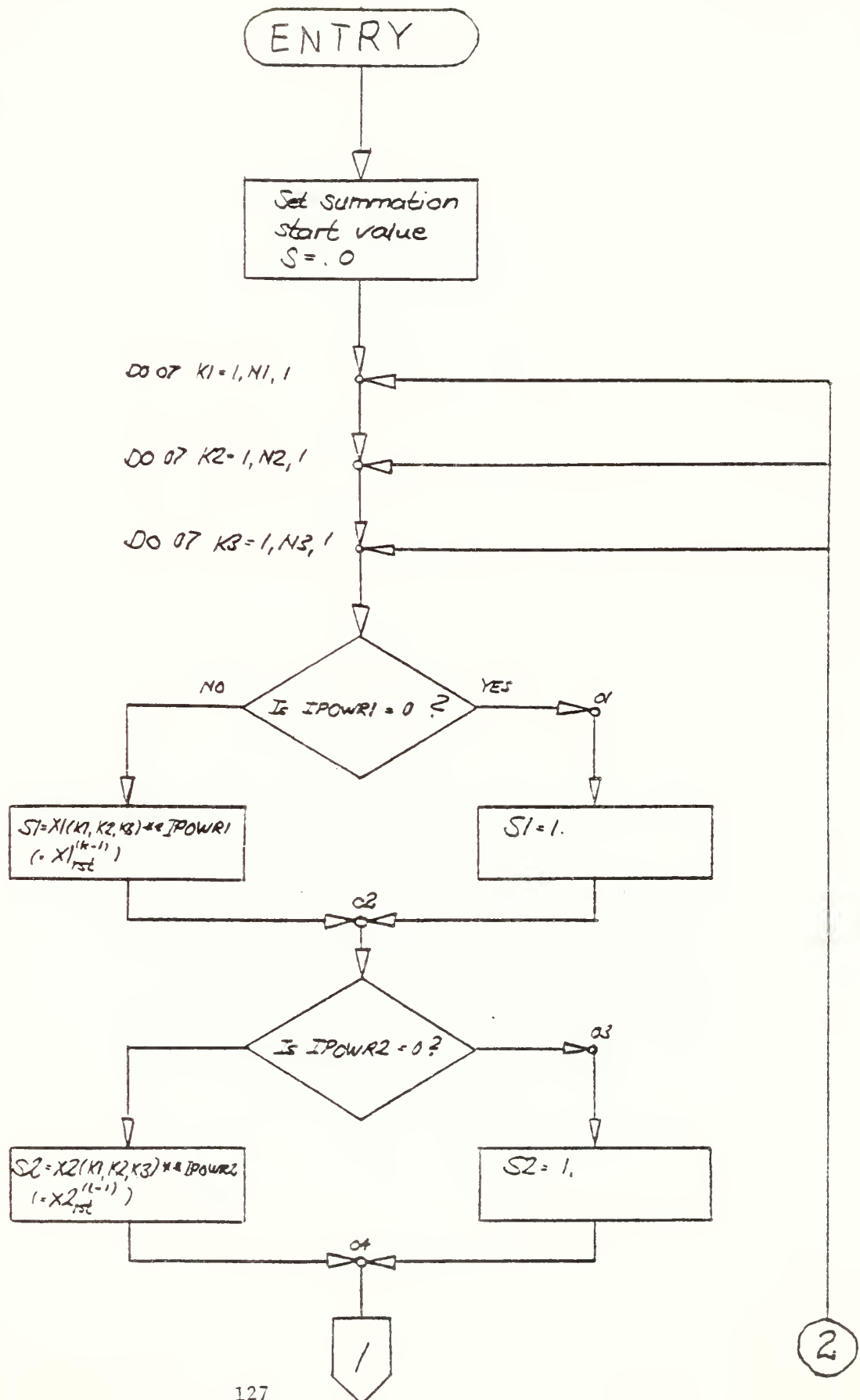
Flow chart MAT42

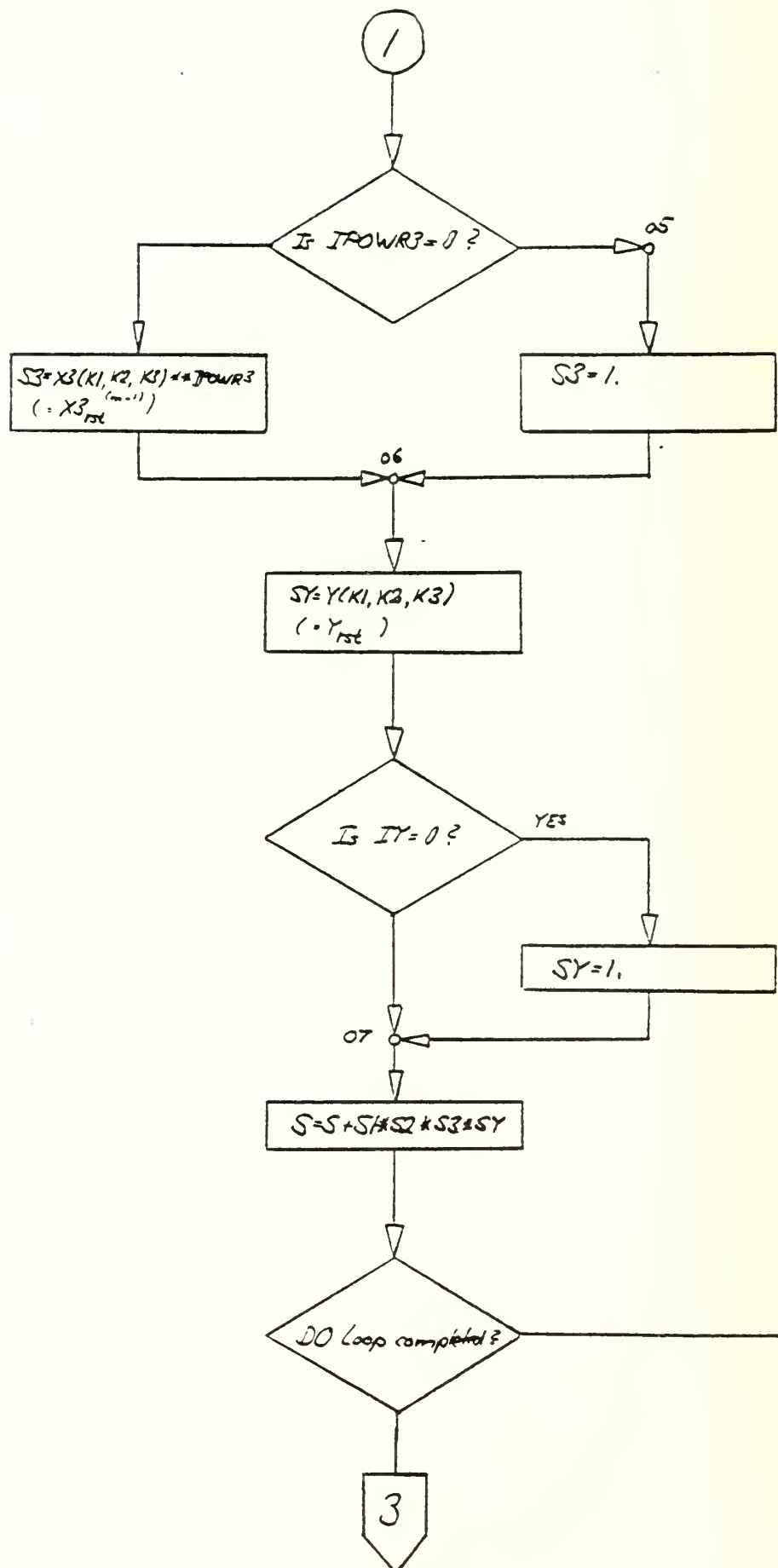


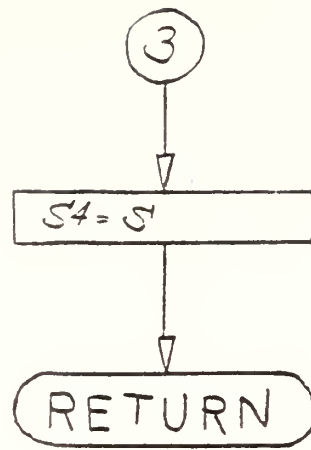




Flow chart S4







APPENDIX C: Software Description: Listings

The following are given:

<u>Listing</u>	<u>Page</u>
<u>Two-Dimensional Approximation</u>	132
BLOCK DATA AB2	132
BLOCK DATA DTA2	133
PROGRAM DEMO2	134
SUBROUTINE MAT2	137
REAL FUNCTION S2	140
INTEGER FUNCTION IEL22	141
SUBROUTINE IEL2	142
REAL FUNCTION F2	145
<u>Three-Dimensional Approximation</u>	146
BLOCK DATA AB3	146
BLOCK DATA DATA3	147
PROGRAM DEMO3	148
SUBROUTINE MAT3	151
SUBROUTINE MAT31	155
REAL FUNCTION S3	156
INTEGER FUNCTION IEL33	157
SUBROUTINE IEL3	158
REAL FUNCTION F3	161
<u>Four-Dimensional Approximation</u>	162
BLOCK DATA AB4	162
BLOCK DATA DATA4	163

Four-Dimensional Approximation (Cont'd)

PROGRAM DEMO4	164
SUBROUTINE MAT4	168
SUBROUTINE MAT41	172
SUBROUTINE MAT42	173
REAL FUNCTION S4	174
INTEGER FUNCTION IEL44	175
SUBROUTINE IEL4	176
REAL FUNCTION F4	179

BLOCK DATA A82

74/175 OPT=2 P10MP

FTN 4.3+495

60/12/29. 14.42.51

PAGE

1

BLOCK DATA A82
COMMON / A82 / 1,3
REAL A(7,7),B(7)
END

BLOCK DATA DTA2 74/175 DPT=2 P=OMP

FTN 4.000000

50/12/24 14.002.11

4300

1

BLOCK DATA DTA2
COMMON / DTA2 / XL,Y
REAL XL(256),Y(256)
END

```

1      PROGRAM DEMO2 (INPUT,OUTPUT,TAPE1,TAPE0,TAPE9=INPUT)
2      .....
3      : THIS IS A DEMONSTRATION PROGRAM AND IT SHOWS THE CORRECT USE :
4      : OF THE TPL BINARY LIBRARY USING EXAMPLES. :
5      : :
6      : AUTHOR: HANS ZEBMER :
7      : DATE: DECEMBER 24, 1980 :
8      : :
9      .....
10     COMMON / DATA /X1,Y
11     REAL X1(256),Y(256)
12     LOGICAL IPLOT
13     DATA PT /3.141592654/
14     DATA IPLOT /.TRUE./
15
16     C FORMATS DEMO2 START
17     101 FORMAT (///72H "HELL!" THIS IS PROGRAM DEMO2 AND WE'LL SEE HOW I
18     *O USE THE MARVELLOUS/70H TPL BINARY LIBRARY. YOUR INTEREST WILL
19     *BE GREATLY AWARDED BY EASY/75H AND FASTER PROGRAMMING-///)
20     102 FORMAT (22H "PLOTTING IS INITIALIZED.")
21     103 FORMAT (24H "DATA POINTS GENERATED.")
22     104 FORMAT (19X,SHX1MIN,0X,SHX1MAX,7X,4MYMIN,7X,4MYMAX/5X,7HPLTTER,4X
23     *IX,F10.3)/4X,4HUSER,4X(1X,F10.3))
24     105 FORMAT (27H "PLOTTING AND USER ARE DEFINED AND AXES DRAWN.")
25     106 FORMAT (20H "DATA POINTS DRAWN.")
26     107 FORMAT (54H "ENTER ORDER OF POLYNOMIAL (HIGHER), IPRINT AND ISTAT")
27     108 FORMAT (22H "CURVE FIT COMPLETED.")
28     109 FORMAT (18H "CURVE FIT DRAWN.")
29     301 FORMAT (21H/14H)
30     C FORMATS DEMO2 STOP
31
32     .....
33     : I/O ASSIGNMENTS. :
34     : :
35     : LOGGLU(ISSSSN) :
36     : LO=0 :
37     : LP=13 :
38     : WRITE (LT, LC1) :
39     : WRITE (LO, 001) :
40
41     .....
42     : INITIALIZE THE PLOTTER. :
43     : :
44     : 12.11 PLOTTING INITIATION (LP) :
45     : WRITE (LI, 102) :
46
47     .....
48     : VARIATION OF POLYNOMIAL DEGREE. LOOP STARTS HERE. :
49     : :
50     : 12.22.1 :
51     : M1STR=1 :
52     : M1STOP=0 :
53     : DO 06 M1STR=M1STR,M1STOP,1 :
54
55     .....
56     : GENERATE DATA SCATTERED AROUND A POLYNOMIAL. :
57     : :
58     : 12.33.1 :
59     : XSTOP = 2.0 :
60     : NPNTS1 = 51 :
61     : OX = (XSTOP-XSTART)/(NPNTS1-1) :
62     : M1STR=1 :
63     : M1STOP=4 :
64     : COEFF(1) = 1.0 :
65     : COEFF(2) = -1.5 :
66     : COEFF(3) = -0.00034 :
67     : COEFF(4) = .3 :
68
69     .....

```

```

COEF(5) = .059
ARG = (2.*PI)/.03125
DO 01 I=1,NPTS1,1
  X(I) = I*START*(I-1)*DX
  Y(I) = FZ(COEF,X,NORORI) + .25*SIN(X/ARG)
01 CONTINUE
  ISTOP1 = NPTS1-1
DO 02 I=1,ISTOP1,2
  DUMMY = Y(I+1)
02 Y(I+1) = Y(I)
  DUMMY = DUMMY
WRITE (LI, 103)

```

C *****

```

: DEFINE PLOTTER AND USER AREAS: DRAW AXES.
:
:

```

C *****

```

XPMIN = 3.
XPMAX = 20.
YPMIN = 2.
YPMAX = 20.
XUMIN = -6.
XUMAX = 2.
YUMIN = -5.
YUMAX = 10.
WRITE (LI, 104) XPMIN,XPMAX,YPMIN,YPMAX,XUMIN,XUMAX,YUMIN,YUMAX
ALPHA = 0.
ALPHA = 90.
XA = (XPMAX - XPMIN) / (XUMAX - YUMIN)
XB = (XPMIN - XPMAX) / (XUMAX - YUMIN)
XL = (XPMAX - YPMIN) / (YUMAX - YUMIN)
YA = (YPMAX - YPMIN) / (YUMAX - YUMIN)
YB = (YPMIN - YPMAX) / (YUMAX - YUMIN)
YL = (YPMAX - YPMIN) / (YUMAX - YUMIN)
IF (IPLOT) CALL AXES (XPMIN,YPMIN,XL,ALPHA,ZH1,2,XUMIN,XUMAX,ZH
* 4,2,2,2)
IF (IPLOT) CALL AXES (XPMIN,YPMIN,YL,ALPHA,ZH Y,2,YUMIN,YUMAX,ZH
* 12,3,3)
WRITE (LI, 105)

```

C *****

```

: PLOT DATA POINTS.
:
:

```

C *****

```

DO 03 I=1,NPTS1,1
  XPLOT = X(I)
  YPLOT = Y(I)
  XPLOT = XPLOT*YA+YB
  YPLOT = YPLOT*XB+YL
03 IF (IPLOT) CALL PLOT (XPLOT,YPLOT,3)
  IF (IPLOT) CALL SYMBL (2,XPLOT,YPLOT)
WRITE (LI, 106)

```

C *****

```

: CALCULATE CURVE FIT THROUGH DATA POINTS.
:
:

```

C *****

```

04 WRITE (LI, 107)
  READ (9,*) NORORI,IPRINT,ISTAT
  NORORI = NORORI
  IPRINT = 2
  ISTAT = 2
  L = NORORI+1
  IF (L.LT.1 .OR. L.GT.7) GO TO 74
  CALL F42 (NPTS1,L,COEF,IPRINT,ISTAT,IERROR)
  IF (IERROR.LT.0) GO TO 06
  WRITE (LI, 108)

```

C *****

```

: PLOT POLYNOMIAL FIT.
:
:

```

C *****


```
00 05 I1=1,NPNTS1,1
      XPLT = I1(I1)
      YAPP = F2(CJEF,I1(I1),NJR041)
      XPLT = XPLT*1048
      YPLT = YAPP*YA*Y9
175  IF (I1.EQ.1 .AND. IPLST) CALL PLIT (XPLT,YPLT,3)
      IF (I1.GT.1 .AND. IPLST) CALL PLIT (XPLT,YPLT,2)
      WRITE (LI,109)
180  IF (IPLST) CALL NUPAGE

      .....
185  : TERMINATE GRAPHICS.
      : .....
      IF (IPLST) CALL STOPG

190  STOP G077
      END
```

```

1 SUBROUTINE MAT2 (NPNTSL,L,COEF,IPRINT,ISTAT,ERROR)
.....
5 PRESET SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B FOR A
20-POLYNOMIAL APPROXIMATION. THE LINEAR EQUATION SYSTEM THEN
IS SOLVED BY USING A GAUSS JORDAN ELIMINATION.
.....
10 AUTHOR: HANS ZEMMER
DATE: DECEMBER 24, 1980
.....
15 VARIABLE EXPLANATION TYPE
NPNTSL** NUMBER OF DATA POINTS INTEGER
L** NUMBER OF COEFFICIENTS FOR THE APPROXIMATING POLYNOMIAL (ORDER+1) INTEGER
COEF(7) POLYNOMIAL COEFFICIENTS REAL
IPRINT** CONTROL QUANTITY OF PRINT OUT INTEGER
*2 PRINT EQUATION SYSTEM BEFORE AND AFTER THE GAUSS JORDAN ELIMINATION.
*1 PRINT EQUATION SYSTEM AFTER THE GAUSS JORDAN ELIMINATION.
*0 PRINT BASIC EQUATIONS WITH ACTUAL PARAMETERS.
*0 NO OUTPUT AT ALL.
20 ISTAT** CONTROL STATISTICS TO BE PRINTED INTEGER
/ISTAT=2 PRINT COORDINATES OF DATA AND APPROXIMATED POINTS AND DEVIATIONS (SE PROP).
/ISTAT=1 ONLY PRINT PROPS.
/ISTAT=0 GET ABSOLUTE ERROR.
/ISTAT<0 GET RELATIVE ERROR.
30 ERROR** ERROR FLAG RETURNED TO THE CALLING PROGRAM. INTEGER
*1 NO ERROR DETECTED. GOOD JOB.
*2 AN ERROR OCCURRED IN THE GAUSSIAN ELIMINATION: COEFFICIENTS ARE NOT ACCURATE. REVIEW PARAMETERS.
.....
40 X(256) X-COORDINATES OF THE DATA POINTS REAL
Y(256) Y-COORDINATES OF THE DATA POINTS REAL
.....
45 IMPORTANT: THE DATA POINTS X AND Y HAVE TO BE PROVIDED THRU
THE LABELED COMMON BLOCK DTA2 AND THE SYSTEM
MATRIX A AND THE RIGHT HAND SIDE VECTOR B ARE
AVAILABLE THROUGH THE LABELED COMMON BLOCK A82.
THIS SAVES A LOT OF THAT PRECIOUS THING CALLED
MEMORY.
.....
50 COMMON / DTA2 / X,Y
COMMON / A82 / A,B
.....
55 REAL X(256),Y(256)
REAL A(7,7),B(7)
REAL COEF(7),STAT(256)
.....
60 C FORMATS MAT2 START
101 FORMAT (///7M AFTER, 13,4TH STEPS NO SOLUTION WITHIN THE ACCURACY
*OF 10.3E-11M 1 (QUIT.//)
601 FORMAT (///51M ORDER OF APPROXIMATION POLYNOMIAL IN X DIRECTION,
*2X,13/40M NUMBER OF DATA POINTS WITH CONSTANT 1.0E-13//)
65 602 FORMAT (///62M SYSTEM MATRIX A AND VECTOR B BEFORE GAUSS JORDAN E
*LIMINATION//)
603 FORMAT ((4X,13(3X,12)))
604 FORMAT (2X,12,13(1X,10E9,2)/(14X,13(1X,10E9,2)))
70 605 FORMAT (///40M EQUATION SYSTEM AFTER GAUSS JORDAN ELIMINATION//)
606 FORMAT (///22M COEFFICIENTS COEF(7)//)
607 FORMAT (2X,3H1 *1(21X,10E9,2)
608 FORMAT (///39M COORDINATES OF DATA POINTS - AS GIVEN//)
609 FORMAT ((5(2X,13,1X,10E9,2)))
75 610 FORMAT (///40M COORDINATES OF DATA POINTS - AS APPROXIMATED//)
611 FORMAT (///30M ABSOLUTE ERROR AT EACH POINT//)
612 FORMAT (///30M RELATIVE ERROR AT EACH POINT IN %//)
613 FORMAT ((5(2X,13,1X,10E9,2)))
614 FORMAT (///21M STATISTICS SUMMARY:1/7X,13MAVERAGE ERROR,10X,10E9.
*27X,19MAXIMUM ERROR AT 1.0E-13,1X,10E9.2)
80 615 FORMAT (///21M STATISTICS SUMMARY:1/7X,13MAVERAGE ERROR,10X,20E9.
*22X,15MAXIMUM ERROR AT 1.0E-13,1X,20E9.2)
616 FORMAT (///7M AFTER, 13,4TH STEPS NO SOLUTION WITHIN THE ACCURACY
*OF 10.3E-11M 1 (QUIT.//)
85 C FORMATS MAT2 STOP

```

```

      LI=LOGLU(ISSSH)
      LU=0
      ERROR=1
      ITRMAX=50
      EPSLON=.100E-007
90      PRINT APPROXIMATION PARAMETERS, IF IPRINT IS GREATER THAN 0.
      NURORI=L-1
95      IF (IPRINT.LT.0) GO TO 01
      WRITE (LO, 601) NURORI, NPNTS1
      START TO PRESET EDGE SECTION ELEMENTS.
100     01 IPWR1=0
      LI=2*L-1
      DO 04 K=1, LI, 1
      IF (K.GT.L) GO TO 02
105     TOP EDGE SECTION ELEMENTS.
      I=1
      J=K
      GO TO 03
110     RIGHT HAND EDGE SECTION ELEMENTS.
      02 J=(K+1)-L
      J=L
115     PRESET EDGE SECTION ELEMENTS IN SYSTEM MATRIX USING S2.
      03 A(I,J)=S2(NPNTS1, IPWR1, 0)
      04 IPWR1=IPWR1+1
120     COPY DEFINED ELEMENTS DIGNALLY.
      LI=L-1
      DO 05 I=1, LI, 1
      DO 05 J=2, L, 1
125     05 A(I+1, J-1)=A(I, J)
      PRESET RIGHT HAND SIDE VECTOR S.
      DO 06 I=1, L, 1
      IPWR1=I-1
130     06 S(I)=S2(NPNTS1, IPWR1, 1)
      PRINT SYSTEM MATRIX AND RIGHT HAND SIDE VECTOR S, IF (IPRINT
135     IS SET TO 2.
      IF (IPRINT.NE.2) GO TO 09
      WRITE (LO, 602)
      WRITE (LO, 603) (J,J=1,L,1)
      DO 07 I=1, L, 1
140     07 WRITE (LO, 604) (A(I,J), J=1, L, 1), 3(I)
      WRITE (LO, 603) (J,J=1,L,1)
      IPRINT=1
145     GAUSS JORDAN ELIMINATION.
      08 NI=L
      EPSLON=EPSLON*41
      CALL IEL2 (NI, ITER, ITRMAX, EPSLON)
150     IF (ITER.LT.0) GO TO 21
      PRINT EQUATION SYSTEM, IF IPRINT IS SET TO 1.
      IF (IPRINT.NE.1) GO TO 10
155     WRITE (LO, 605)
      WRITE (LO, 603) (J,J=1,L,1)
      DO 09 I=1, L, 1
      09 WRITE (LO, 606) (A(I,J), J=1, L, 1), 3(I)
      WRITE (LO, 603) (J,J=1,L,1)
160     DEFINE COEFFICIENTS AND PRINT THEM, IF IPRINT IS GREATER THAN 0.
      10 CONTINUE
      DO 11 LI=L, L, 1
165     11 COEF(LI)=9(LI)
      IF (IPRINT.LT.0) GO TO 13
      WRITE (LO, 606)
      DO 12 LI=L, L, 1
170     12 WRITE (LO, 607) LI, C7EF(LI)

```

```

C      STATISTICS, IF ISTAT ANYTHING ELSE BUT 0.
13 IF (ISTAT.EQ.0) GO TO 23
175 IF (ISTAT.LT.0) IERR=1
    IF (ISTAT.GT.0) IERR=2
    ISTAT=ABS(ISTAT)
C      COMPUTE ABSOLUTE ERROR.
180 I=(APPROXIMATED VALUE - GIVEN VALUE)
C      DO 14 I=1,NPNTS,1
14 STAT(I)=F2(COEF,X1(I),NORDR1)-Y(I)
C      COMPUTE RELATIVE ERRJR, IF ISTAT WAS LESS THAN ZERO, I.E. IERR=1.
185 I=(APPROXIMATED VALUE - GIVEN VALUE)/(GIVEN VALUE)
    IF (IERROR.NE.1) GO TO 16
    DO 15 I=1,NPNTS,1
190 STAT(I)=STAT(I)/Y(I)
C      ERROR STATISTICS: COMPUTE AVERAGE ERROR AND FIND LOCATION
    OF MAXIMUM ERROR.
195 ERRAVE=0
    ERRMAX=ABS(STAT(1))
    IMAX=1
    DO 17 I=1,NPNTS,1
    IF (ABS(STAT(I)).LE.ERRMAX) GO TO 17
    ERRMAX=ABS(STAT(I))
200 IMAX=I
17 ERRAVE=ERRAVE+ABS(STAT(I))
    ERRAVE=ERRAVE/NPNTS
    IF (ISTAT.NE.2) GO TO 19
C      PRINT COORDINATES OF DATA POINTS AND APPROXIMATED VALUES,
    IF /ISTAT/ IS EQUAL 2.
205 WRITE (LO, 005)
    WRITE (LO, 009) (I,Y(I),I=1,NPNTS,1)
    WRITE (LO, 010)
    WRITE (LO, 009) (I,F2(COEF,X1(I),NORDR1),I=1,NPNTS,1)
    ISTAT=1
210 PRINT ERRORS, IF /ISTAT/ WAS 1 OR 2.
215 IF (ISTAT.NE.1) GO TO 20
    IF (IERROR.EQ.1) GO TO 19
C      PRINT ABSOLUTE ERRORS.
220 WRITE (LO, 011)
    WRITE (LO, 009) (I,STAT(I),I=1,NPNTS,1)
    GO TO 20
225 PRINT RELATIVE ERRORS.
19 WRITE (LO, 012)
    WRITE (LO, 013) (I,STAT(I),I=1,NPNTS,1)
230 PRINT ERROR STATISTICS.
20 IF (IERROR.EQ.2) WRITE (LO, 014) ERRAVE,IMAX,ERRMAX
    IF (IERROR.EQ.1) WRITE (LO, 013) ERRAVE,IMAX,ERRMAX
    GO TO 23
235 FATAL ERROR DETECTED. OUTPUT ERROR MESSAGE AND PRESET COEFFICIENTS.
21 ITER=ITER+1
    IERR=IERR+1
    B(1)=Y(1)
    DO 22 IC=2,N+1
22 B(IC)=0
    WRITE (LO, 001) ITER,EPSLON
    WRITE (LO, 015) ITER,EPSLON
    GO TO 10
245 RETURN.
23 RETURN
250 END

```

```

1  REAL FUNCTION S2 (NPNTS1,IPWR1,Y)
2  .....
3  * PERFORM SUMMATIONS TO PROVIDE THE COEFFICIENTS A(1:4) IN THE
4  * SYSTEM MATRIX A AND THE ELEMENTS OF THE RIGHT HAND SIDE VECTOR B.
5  * THE COEFFICIENTS A(1:4) ARE CALCULATED BY THE FOLLOWING EQUATIONS:
6  * BE SOLVED FOR A POLYNOMIAL CO-APPROXIMATION.
7  .....
8  *
9  *
10 * AUTHOR: JAMES T. SMITH
11 * DATE: DECEMBER 24, 1980
12 *
13 * VARIABLE EXPLANATION TYPE
14 * NPNTS1 NUMBER OF DATA POINTS, CONSTANT 11 INTEGER
15 * IPWR1 POWER OF THE XI VALUES IN THE SUMMATION, CONSTANT 12 INTEGER
16 * IT CONTROL PARAMETER, 10 Y VALUES ARE NOT INCLUDED IN THE SUMMATION, 11 Y VALUES ARE INCLUDED IN THE SUMMATION, 12 Y VALUES ARE INCLUDED IN THE SUMMATION. INTEGER
17 * S2 RESULT OF SUMMATION, REAL
18 *
19 * XI(256) XI-COORDINATES OF THE DATA POINTS REAL
20 * Y(256) Y-COORDINATES OF THE DATA POINTS REAL
21 *
22 * IMPORTANT: THE DATA POINTS XI AND Y HAVE TO BE PROVIDED THRU
23 * THE LABELED COMMON BLOCK DPA2.
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197 *
198 *
199 *
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *
1001 *
1002 *
1003 *
1004 *
1005 *
1006 *
1007 *
1008 *
1009 *
1010 *
1011 *
1012 *
1013 *
1014 *
1015 *
1016 *
1017 *
1018 *
1019 *
1020 *
1021 *
1022 *
1023 *
1024 *
1025 *
1026 *
1027 *
1028 *
1029 *
1030 *
1031 *
1032 *
1033 *
1034 *
1035 *
1036 *
1037 *
1038 *
1039 *
1040 *
1041 *
1042 *
1043 *
1044 *
1045 *
1046 *
1047 *
1048 *
1049 *
1050 *
1051 *
1052 *
1053 *
1054 *
1055 *
1056 *
1057 *
1058 *
1059 *
1060 *
1061 *
1062 *
1063 *
1064 *
1065 *
1066 *
1067 *
1068 *
1069 *
1070 *
1071 *
1072 *
1073 *
1074 *
1075 *
1076 *
1077 *
1078 *
1079 *
1080 *
1081 *
1082 *
1083 *
1084 *
1085 *
1086 *
1087 *
1088 *
1089 *
1090 *
1091 *
1092 *
1093 *
1094 *
1095 *
1096 *
1097 *
1098 *
1099 *
1100 *
1101 *
1102 *
1103 *
1104 *
1105 *
1106 *
1107 *
1108 *
1109 *
1110 *
1111 *
1112 *
1113 *
1114 *
1115 *
1116 *
1117 *
1118 *
1119 *
1120 *
1121 *
1122 *
1123 *
1124 *
1125 *
1126 *
1127 *
1128 *
1129 *
1130 *
1131 *
1132 *
1133 *
1134 *
1135 *
1136 *
1137 *
1138 *
1139 *
1140 *
1141 *
1142 *
1143 *
1144 *
1145 *
1146 *
1147 *
1148 *
1149 *
1150 *
1151 *
1152 *
1153 *
1154 *
1155 *
1156 *
1157 *
1158 *
1159 *
1160 *
1161 *
1162 *
1163 *
1164 *
1165 *
1166 *
1167 *
1168 *
1169 *
1170 *
1171 *
1172 *
1173 *
1174 *
1175 *
1176 *
1177 *
1178 *
1179 *
1180 *
1181 *
1182 *
1183 *
1184 *
1185 *
1186 *
1187 *
1188 *
1189 *
1190 *
1191 *
1192 *
1193 *
1194 *
1195 *
1196 *
1197 *
1198 *
1199 *
1200 *
1201 *
1202 *
1203 *
1204 *
1205 *
1206 *
1207 *
1208 *
1209 *
1210 *
1211 *
1212 *
1213 *
1214 *
1215 *
1216 *
1217 *
1218 *
1219 *
1220 *
1221 *
1222 *
1223 *
1224 *
1225 *
1226 *
1227 *
1228 *
1229 *
1230 *
1231 *
1232 *
1233 *
1234 *
1235 *
1236 *
1237 *
1238 *
1239 *
1240 *
1241 *
1242 *
1243 *
1244 *
1245 *
1246 *
1247 *
1248 *
1249 *
1250 *
1251 *
1252 *
1253 *
1254 *
1255 *
1256 *
1257 *
1258 *
1259 *
1260 *
1261 *
1262 *
1263 *
1264 *
1265 *
1266 *
1267 *
1268 *
1269 *
1270 *
1271 *
1272 *
1273 *
1274 *
1275 *
1276 *
1277 *
1278 *
1279 *
1280 *
1281 *
1282 *
1283 *
1284 *
1285 *
1286 *
1287 *
1288 *
1289 *
1290 *
1291 *
1292 *
1293 *
1294 *
1295 *
1296 *
1297 *
1298 *
1299 *
1300 *
1301 *
1302 *
1303 *
1304 *
1305 *
1306 *
1307 *
1308 *
1309 *
1310 *
1311 *
1312 *
1313 *
1314 *
1315 *
1316 *
1317 *
1318 *
1319 *
1320 *
1321 *
1322 *
1323 *
1324 *
1325 *
1326 *
1327 *
1328 *
1329 *
1330 *
1331 *
1332 *
1333 *
1334 *
1335 *
1336 *
1337 *
1338 *
1339 *
1340 *
1341 *
1342 *
1343 *
1344 *
1345 *
1346 *
1347 *
1348 *
1349 *
1350 *
1351 *
1352 *
1353 *
1354 *
1355 *
1356 *
1357 *
1358 *
1359 *
1360 *
1361 *
1362 *
1363 *
1364 *
1365 *
1366 *
1367 *
1368 *
1369 *
1370 *
1371 *
1372 *
1373 *
1374 *
1375 *
1376 *
1377 *
1378 *
1379 *
1380 *
1381 *
1382 *
1383 *
1384 *
1385 *
1386 *
1387 *
1388 *
1389 *
1390 *
1391 *
1392 *
1393 *
1394 *
1395 *
1396 *
1397 *
1398 *
1399 *
1400 *
1401 *
1402 *
1403 *
1404 *
1405 *
1406 *
1407 *
1408 *
1409 *
1410 *
1411 *
1412 *
1413 *
1414 *
1415 *
1416 *
1417 *
1418 *
1419 *
1420 *
1421 *
1422 *
1423 *
1424 *
1425 *
1426 *
1427 *
1428 *
1429 *
1430 *
1431 *
1432 *
1433 *
1434 *
1435 *
1436 *
1437 *
1438 *
1439 *
1440 *
1441 *
1442 *
1443 *
1444 *
1445 *
1446 *
1447 *
1448 *
1449 *
1450 *
1451 *
1452 *
1453 *
1454 *
1455 *
1456 *
1457 *
1458 *
1459 *
1460 *
1461 *
1462 *
1463 *
1464 *
1465 *
1466 *
1467 *
1468 *
1469 *
1470 *
1471 *
1472 *
1473 *
1474 *
1475 *
1476 *
1477 *
1478 *
1479 *
1480 *
1481 *
1482 *
1483 *
1484 *
1485 *
1486 *
1487 *
1488 *
1489 *
1490 *
1491 *
1492 *
1493 *
1494 *
1495 *
1496 *
1497 *
1498 *
1499 *
1500 *
1501 *
1502 *
1503 *
1504 *
1505 *
1506 *
1507 *
1508 *
1509 *
1510 *
1511 *
1512 *
1513 *
1514 *
1515 *
1516 *
1517 *
1518 *
1519 *
1520 *
1521 *
1522 *
1523 *
1524 *
1525 *
1526 *
1527 *
1528 *
1529 *
1530 *
1531 *
1532 *
1533 *
1534 *
1535 *
1536 *
1537 *
1538 *
1539 *
1540 *
1541 *
1542 *
1543 *
1544 *
1545 *
1546 *
1547 *
1548 *
1549 *
1550 *
1551 *
1552 *
1553 *
1554 *
1555 *
1556 *
1557 *
1558 *
1559 *
1560 *
1561 *
1562 *
1563 *
1564 *
1565 *
1566 *
1567 *
1568 *
1569 *
1570 *
1571 *
1572 *
1573 *
1574 *
1575 *
1576 *
1577 *
1578 *
1579 *
1580 *
1581 *
1582 *
1583 *
1584 *
1585 *
1586 *
1587 *
1588 *
1589 *
1590 *
1591 *
1592 *
1593 *
1594 *
1595 *
1596 *
1597 *
1598 *
1599 *
1600 *
1601 *
1602 *
1603 *
1604 *
1605 *
1606 *
1607 *
1608 *
1609 *
1610 *
1611 *
1612 *
1613 *
1614 *
1615 *
1616 *
1617 *
1618 *
1619 *
1620 *
1621 *
1622 *
1623 *
1624 *
1625 *
1626 *
1627 *
1628 *
1629 *
1630 *
1631 *
1632 *
1633 *
1634 *
1635 *
1636 *
1637 *
1638 *
1639 *
1640 *
1641 *
1642 *
1643 *
1644 *
1645 *
1646 *
1647 *
1648 *
1649 *
1650 *
1651 *
1652 *
1653 *
1654 *
1655 *
1656 *
1657 *
1658 *
1659 *
1660 *
1661 *
1662 *
1663 *
1664 *
1665 *
1666 *
1667 *
1668 *
1669 *
1670 *
1671 *
1672 *
1673 *
1674 *
1675 *
1676 *
1677 *
1678 *
1679 *
1680 *
1681 *
1682 *
1683 *
1684 *
1685 *
1686 *
1687 *
1688 *
1689 *
1690 *
1691 *
1692 *
1693 *
1694 *
1695 *
1696 *
1697 *
1698 *
1699 *
1700 *
1701 *
1702 *
1703 *
1704 *
1705 *
1706 *
1707 *
1708 *
1709 *
1710 *
1711 *
1712 *
1713 *
1714 *
1715 *
1716 *
1717 *
1718 *
1719 *
1720 *
1721 *
1722 *
1723 *
1724 *
1725 *
1726 *
1727 *
1728 *
1729 *
1730 *
1731 *
1732 *
1733 *
1734 *
1735 *
1736 *
1737 *
1738 *
1739 *
1740 *
1741 *
1742 *
1743 *
1744 *
1745 *
1746 *
1747 *
1748 *
1749 *
1750 *
1751 *
1752 *
1753 *
1754 *
1755 *
1756 *
1757 *
1758 *
1759 *
1760 *
1761 *
1762 *
1763 *
1764 *
1765 *
1766 *
1767 *
1768 *
1769 *
1770 *
1771 *
1772 *
1773 *
1774 *
1775 *
1776 *
1777 *
1778 *
1779 *
1780 *
1781 *
1782 *
1783 *
1784 *
1785 *
1786 *
1787 *
1788 *
1789 *
1790 *
1791 *
1792 *
1793 *
1794 *
1795 *
1796 *
1797 *
1798 *
1799 *
1800 *
1801 *
1802 *
1803 *
1804 *
1805 *
1806 *
1807 *
1808 *
1809 *
1810 *
1811 *
1812 *
1813 *
1814 *
1815 *
1816 *
1817 *
1818 *
1819 *
1820 *
1821 *
1822 *
1823 *
1824 *
1825 *
1826 *
1827 *
1828 *
1829 *
1830 *
1831 *
1832 *
1833 *
1834 *
1835 *
1836 *
1837 *
1838 *
1839 *
1840 *
1841 *
1842 *
1843 *
1844 *
1845 *
1846 *
1847 *
1848 *
1849 *
1850 *
1851 *
1852 *
1853 *
1854 *
1855 *
1856 *
1857 *
1858 *
1859 *
1860 *
1861 *
1862 *
1863 *
1864 *
1865 *
1866 *
1867 *
1868 *
1869 *
1870 *
1871 *
1872 *
1873 *
1874 *
1875 *
1876 *
1877 *
1878 *
1879 *
1880 *
1881 *
1882 *
1883 *
1884 *
1885 *
1886 *
1887 *
1888 *
1889 *
1890 *
1891 *
1892 *
1893 *
1894 *
1895 *
1896 *
1897 *
1898 *
1899 *
1900 *
1901 *
1902 *
1903 *
1904 *
1905 *
1906 *
1907 *
1908 *
1909 *
1910 *
1911 *
1912 *
1913 *
1914 *
1915 *
1916 *
1917 *
1918 *
1919 *
1920 *
1921 *
1922 *
1923 *
1924 *
1925 *
1926 *
1927 *
1928 *
1929 *
1930 *
1931 *
1932 *
1933 *
1934 *
1935 *
1936 *
1937 *
1938 *
1939 *
1940 *
1941 *
1942 *
1943 *
1944 *
1945 *
1946 *
1947 *
1948 *
1949 *
1950 *
1951 *
1952 *
1953 *
1954 *
1955 *
1956 *
1957 *
1958 *
1959 *
1960 *
1961 *
1962 *
1963 *
1964 *
1965 *
1966 *
1967 *
1968 *
1969 *
1970 *
1971 *
1972 *
1973 *
1974 *
1975 *
1976 *
1977 *
1978 *
1979 *
1980 *
1981 *
1982 *
1983 *
1984 *
1985 *
1986 *
1987 *
1988 *
1989 *
1990 *
1991 *
1992 *
1993 *
1994 *
1995 *
1996 *
1997 *
1998 *
1999 *
2000 *
2001 *
2002 *
2003 *
2004 *
2005 *
2006 *
2007 *
2008 *
2009 *
2010 *
2011 *
2012 *
2013 *
2014 *
2015 *
2016 *
2017 *
2018 *
2019 *
2020 *
2021 *
2022 *
2023 *
2024 *
2025 *
2026 *
2027 *
2028 *
2029 *
2030 *
2031 *
2032 *
2033 *
2034 *
2035 *
2036 *
2037 *
2038 *
2039 *
2040 *
2041 *
2042 *
2043 *
2044 *
2045 *
2046 *
2047 *
2048 *
2049 *
2050 *
2051 *
2052 *
2053 *
2054 *
2055 *
2056 *
2057 *
2058 *
2059 *
2060 *
2061 *
2062 *
2063 *
2064 *
2065 *
2066 *
2067 *
2068 *
2069 *
2070 *
2071 *
2072 *
2073 *
2074 *
2075 *
2076 *
2077 *
2078 *
2079 *
2080 *
2081 *
2082 *
2083 *
2084 *
2085 *
2086 *
2087 *
2088 *
2089 *
2090 *
2091 *
2092 *
2093 *
2094 *
2095 *
2096 *
2097 *
2098 *
2099 *
2100 *
2101 *
2102 *
2103 *
2104 *
2105 *
2106 *
2107 *
2108 *
2109 *
2110 *
2111 *
2112 *
2113 *
2114 *
2115 *
2116 *
2117 *
2118 *
2119 *
2120 *
2121 *
2122 *
2123 *
2124 *
2125 *
2126 *
2127 *
2128 *
2129 *
2130 *
2131 *
2132 *
2133 *
2134 *
2135 *
2136 *
2137 *
2138 *
2139 *
2140 *
2141 *
2142 *
2143 *
2144 *
2145 *
2146 *
2147 *
2148 *
2149 *
2150 *
2151 *
2152 *
2153 *
2154 *
2155 *
2156 *
2157 *
2158 *
2159 *
2160 *
2161 *
2162 *
2163 *
2164 *
2165 *
2166 *
2167 *
2168 *
2169 *
21
```

1 INTEGER FUNCTION IEL22 (M)

.....
 5 GAUSS JORDAN ELIMINATION WITH MATRIX A AND VECTOR B. THE
 VARIABLE IEL22 IS SET TO 1 IF THE ELIMINATION IS COMPLETED.
 IEL22 LESS THAN 0 INDICATES AN ERROR.

10 AUTHOR: JAMES ZERNER
 DATE: DECEMBER 24, 1980
 VARIABLE EXPLANATION TYPE
 M..... NUMBER OF EQUATIONS..... INTEGER
 IEL2 CONTROL PARAMETER, RETURNED TO THE CALLING ROUTINE. INTEGER
 15 *1 GAUSS JORDAN ELIMINATION WAS PERFORMED CORRECTLY.
 *0 ERROR ENCOUNTERED DURING GAUSS JORDAN ELIMINATION.

20 A(7,7) SYSTEM MATRIX A REAL
 B(7) RIGHT HAND SIDE VECTOR B REAL

25 IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
 THE LABELED COMMON BLOCK ABZ. BOTH A AND B WILL
 UNDERGO CHANGES DURING EXECUTION, I.E. THE
 ELIMINATION.

30 COMMON / ABZ / A,B

REAL A(7,7),B(7)

35 IEL2=1
 IEL22=IEL2
 EPS=1.E-030
 DO 01 I=1,M+1
 DO 02 K=1,M+1
 IF (K.EQ.I) GO TO 02
 CONST=-A(K,I)/A(I,I)
 DO 01 J=1,M+1
 A(K,J)=A(K,J)+CONST*A(I,J)
 IF (J.EQ.I) A(K,J)=0
 01 CONTINUE
 B(K)=B(K)+CONST*B(I)
 02 CONTINUE
 DO 03 J=1,M+1
 A(I,J)=A(I,J)/CONST
 A(I,I)=1
 B(I)=B(I)/CONST
 DO 04 J=1,M+1
 ERROR=0
 DO 04 K=1,M+1
 IF (ABS(A(J,K)).GT.EPS) GO TO 04
 IERROR=IERROR+1
 04 CONTINUE
 IF (ABS(B(I)).GT.EPS) GO TO 05
 IERROR=IERROR+1
 05 IF (IERROR.GT.M+1) GO TO 07
 06 CONTINUE
 GO TO 08
 07 IEL2=0
 08 RETURN
 IEL22=IEL2
 ENO

```

1      SUBROUTINE IEL2 (N,ITER,ITMAX,EPSLON)
2      .....
3      GAUSSIAN ELIMINATION WITH LINEAR EQUATION SYSTEM V WITH RIGHT
4      HAND SIDE VECTOR B IS INTEGRATED TOGETHER WITH THE SYSTEM A=
5      TRIX A IN THE MATRIX V. ITER IS THE NUMBER OF ITERATIONS
6      TURNED TO THE CALLING PROGRAM. THE ITERATION TO IMPROVE THE
7      SOLUTION SO MANY TIMES, UNTIL THE REMAINING ERROR IS < EPSLON
8      THE MAGNITUDE OF EPSLON SHOULD BE THE NUMBER OF ITERATIONS 24-
9      CEO ITMAX, THE SUBROUTINE TERMINATES THE CALCULATION AND
10     SETS ITER TO A NEGATIVE VALUE.
11
12     AUTHORS: KONRAD VOGELER, HANS ZEBNER
13     DATE: DECEMBER 24, 1980
14
15     VARIABLE EXPLANATION TYPE
16     .....
17     ITER NUMBER OF ITERATIONS TO BE RETURNED INTEGER
18     NO GAUSSIAN ELIMINATION PERFORMED
19     CORRECTLY.
20     NO CONVERGENCE OF THE SOLUTION FOR
21     THE DESIRED RESOLUTION EPSLON.
22     ITERATION TO TERMINATE AFTER
23     ITMAX ITERATION STEPS.
24     ITMAX MAXIMUM NUMBER OF ITERATION STEPS INTEGER
25     EPSLON DESIRED RESOLUTION OF THE SOLUTION REAL
26
27     A(7,7) SYSTEM MATRIX A REAL
28     B(7) RIGHT HAND SIDE VECTOR B REAL
29
30     IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
31     THE LABELED COMMON BLOCK SUBROUTINE A AND B WILL
32     WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE
33     ELIMINATION.
34     .....
35
36     COMMON / A82 / V
37     REAL V(7,8),G(9),A(7,9),B(8)
38     LOGICAL IPRINT
39     DATA IPRINT / .FALSE. /
40     DATA PI / 3.14159284 /
41
42     C FORMATS IEL2 START
43     601 FORMAT (//39H MATRIX A AND VECTOR B FOR ELIMINATION/)
44     602 FORMAT ((4X,13(8X,12)))
45     603 FORMAT (2X,12,13(1X,1PE9.2)) / ((4X,13(1X,1PE9.2)))
46     604 FORMAT (//25H DIVIDES SYSTEM (N,1) BY (ITERATIONS/SCHRIFFT/
47     )
48     605 FORMAT (//2X,12,24X Koeffizienten aus Gauss/)
49     606 FORMAT ((2X,12,1X,1PE9.2))
50     607 FORMAT (//39H Koeffizienten zeilenweise schrittweise (N=6,13,3))
51     608 FORMAT (27X,12H Koeffizienten zeilenweise schrittweise (N=6,13,3))
52     609 FORMAT (//20H 1. Zeile V(1,2) bis V(1,7) = 6,2,3/)
53     610 FORMAT (//25H SYSTEM A nach Elimination/)
54     C FORMATS IEL2 STOP
55
56     LJ=0
57     M=N-1
58     K=N-1
59
60     REARRANGE VECTOR B IN ARRAY V, IF M IS LESS THAN 7.
61
62     IF (M<7) GO TO J2
63     DO 01 I=1,M,1
64     01 V(I,M)=V(I,8)
65
66     PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B BEFORE THE
67     ELIMINATION, IF IPRINT IS SET .TRUE.
68
69     02 IF (IPRINT) WRITE (LJ, 601)
70     IF (IPRINT) WRITE (LJ, 602) ((J,J=1,M,1))
71     DO 03 I=1,M,1
72     03 IF (IPRINT) WRITE (LJ, 603) (V(I,J),J=1,M,1)
73     IF (IPRINT) WRITE (LJ, 602) ((J,J=1,M,1))
74
75     LINEAR EQUATION SYSTEM V(N,1)G(N)=V(N,M+1)
76
77     DO 04 I=1,M,1
78     04 A(I,J)=V(I,J)
79     DO 05 J=1,M,1
80     05 B(I)=V(I,J)
81
82     CHECK FOR STABILITY.

```

C

```

      FI=.0
      DO 06 I=1,M,1
06    G(I)=.0
      DO 08 I=1,M,1
      DO 07 J=1,N,1
07    G(I)=G(I)+V(I,J)**2
08    G(I)=SQRT(G(I))
      DO 10 I=1,K,1
      JJ=I+1
      DO 10 J=JJ,M,1
      SK=.0
      DO 09 I3=1,M,1
09    SK=SK+V(I,I3)*V(J,I3)
      F=SK/(G(I)*G(J))
      IF (ABS(F).GT.ABS(FI)) FI=F
10    CONTINUE
      F=ABS(FI)
      FI=ACOS(FI)
      FI=(FI*180.)/PI

```

```

      START ITERATION TO FIND AND IMPROVE SOLUTION.

```

```

      DO 27 LX=1,999,1
      ITER=LX
      DO 13 J=1,K,1

```

```

      PIVOTISATION AND GAUSSIAN STEP FOR COLUMN J.

```

```

      IX=J
      EV=V(J,J)
      DO 12 I=J,M,1
      IF (ABS(V(I,J)).GT.ABS(EV)) GO TO 11
      GO TO 12
11    IX=I
12    CONTINUE
      IF (IX.EQ.J) GO TO 14
      DO 13 I=J,M,1
      G(I)=V(I,IX)
      V(I,IX)=V(I,J)
13    V(J,IX)=G(I)
      G(M)=.0
      MM=J+1
      DO 15 I=MM,M,1
      WV=V(I,J)/V(J,J)
      DO 15 KK=J,M
      V(I,KK)=V(I,KK)-V(J,KK)*WV
15    CONTINUE
      IF (V(I,I).EQ.0.0) GO TO 32

```

```

      SOLUTION.

```

```

      DO 16 I=1,M,1
16    G(I)=.0
      G(M)=V(M,M)/V(M,M)
      DO 19 I=1,K,1
      I4=I
      IF (V(I,I).EQ.0.0) GO TO 32
      G(I)=V(I,M)
      DO 17 J=1,K,1
      J4=J
      IF (J.EQ.I) GO TO 18
17    G(I)=G(I)-G(J)*V(I,J)
18    CONTINUE
19    G(I)=G(I)/V(I,I)

```

```

      CORRECT SOLUTION BY AMOUNT OF ERROR, WHICH WAS INVOLVED IN THIS
      ITERATION STEP. TAKE THIS VECTOR AS START VECTOR FOR THE NEXT
      ITERATION.

```

```

      DO 20 I=1,M,1
20    G(I)=G(I)+G(I)

```

```

      COMPUTE THE REMAINING ERROR.

```

```

      DO 22 I=1,M,1
      V(I,M)=.0
      DO 21 J=1,M,1
21    V(I,M)=V(I,M)+G(J)*A(I,J)
22    V(I,M)=A(I,M)-V(I,M)
      R=.0
      DO 23 I=1,M,1
23    R=R+V(I,M)*V(I,M)

```



```

      R=SQRT(X)
      IF (R.LT.EPSLON) GO TO 26
      IF (LA.GT.ITERMAX) GO TO 33
      DO 24 I=1,N+1
175      24  S(I)=G(I)
      PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B AFTER THE
      PIVOTISATION.
180      IF (IPRINT) WRITE (LO, 604) LX
      IF (IPRINT) WRITE (LO, 602) (JJ=1,M+1)
      DO 25 I=1,M+1
185      25  IF (IPRINT) WRITE (LO, 603) I,(V(I,J),J=1,M+1),G(I)
      IF (IPRINT) WRITE (LO, 602) (JJ=1,M+1)
      DO 26 I=1,M+1
      DO 26 J=1,M+1
      26  V(I,J)=A(I,J)
      STOP LOOP TO FIND SOLUTION.
190      27 CONTINUE
      REARRANGE SOLUTION IN THE ARRAY V.
199      28 CONTINUE
      DO 29 I=1,M+1
      29  V(I,8)=G(I)
      PRINT SOLUTION, IF IPRINT IS SET .TRUE. .
200      30 IF (IPRINT) WRITE (LO, 605) 4
      IF (IPRINT) WRITE (LO, 605) (I,G(I),I=1,M+1)
      IF (IPRINT) WRITE (LO, 607) 51
      IF (IPRINT) WRITE (LO, 608) 4*LX
205      DRUCKEN DER MATRIX A UND DES VECTORS B NACH ELIMINATION.
      IF (IPRINT) WRITE (LO, 613)
      IF (IPRINT) WRITE (LO, 602) (JJ=1,M+1)
      DO 31 I=1,M+1
210      31 IF (IPRINT) WRITE (LO, 603) I,(V(I,J),J=1,M+1)
      IF (IPRINT) WRITE (LO, 602) (JJ=1,M+1)
      RETURN
215      ERROR ENCOUNTERED. STOP.
      32 ITER=ITER+1
      WRITE (LO, 649) I,I,V(I,I)
      G(I)=G(I)/V(I,I)
      RETURN
220      NO CONVERGENCE OF THE SOLUTION.
      33 ITER=ITER+1
      RETURN
225      END

```

1 REAL FUNCTION F2 (CDEF,X1,N,NORDR1)

5 * COMPUTE FUNCTION VALUE $Y=F(X1)$ OF A 2D-POLYNOMIAL USING
 * THE HORNER SCHEME IN ORDER TO INCREASE SPEED AND ACCURACY.

10 * AUTHOR: HANS ZEBNER
 * DATE: DECEMBER 24, 1980

VARIABLE	EXPLANATION	TYPE
CDEF(I)	2D-POLYNOMIAL COEFFICIENTS	REAL
X1	X1-VALUE, WHERE Y WILL BE COMPUTED.	REAL
NORDR1	ORDER OF THE POLYNOMIAL IN THE X1 DIRECTION.	INTEGER
F2	ORDINATE OF THE 2D-POLYNOMIAL.	REAL

20 REAL COEF(7)
 NCDEF=NORDR1+1
 Y=COEF(NCDEF)
 DO 1 I=1,NORDR1,1
 1 Y=Y*X1+COEF(I)
 F2=Y
 RETURN
 END

BLOCK DATA A83

74/175 JPT=2 P10MP

FTN 9.00000

00/42/27. 17. 9.91

Page 4

1

BLOCK DATA A83
COMMON / A83 / A,3
REAL A(49,49),B(49)
END

BLOCK DATA DATA3 74/175 DPT=2 27044

SIN 4.000000

00/22/24 14.00.00

PAGE

1

1

BLOCK DATA DATA3
COMMON / DATA3 / X1,X2,T
REAL X1(10,10),X2(10,10),T(10,10)
END

```

1      PROGRAM DEMO3 (INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4=INPUT)
2      .....
3      : THIS IS A DEMONSTRATION PROGRAM AND IT SHOWS THE CORRECT USE
4      : OF THE IPL BINARY LIBRARY USING EXAMPLES.
5      :
6      : AUTHOR: JAMES ZEGNER
7      : DATE: DECEMBER 31, 1980
8      :
9      .....
10     COMMON / AFLO / PLOTR
11     COMMON / DATA3 / X1,X2,Y
12
13     REAL PLOTR(256)
14     REAL X1(16,16),X2(16,16),Y(16,16)
15
16     REAL COEF(49),SU3M1(7,7)
17     INTEGER IPW(5)
18     LOGICAL IPLOT
19
20     DATA ICMT /10-10 /
21     DATA IPLOT /.TRUE./
22     DATA PI /3.141592654/
23
24     C FORMATS DEMO3 START
25     101 FORMAT (//72H HELLO - THIS IS PROGRAM DEMO3 AND WE'LL SEE HOW I
26     *USE THE MARVELLOUS/70H IPL BINARY LIBRARY. YOUR INTEREST WILL
27     *BE GREATLY AWARDED BY EASIER/25H AND FASTER PROGRAMMING//)
28     102 FORMAT (22H PLOTTER INITIALIZED)
29     103 FORMAT (31H DATA POINTS READ FROM TAPE 0.)
30     104 FORMAT (14H 5X11XIN,0X,1X11XAX,0X,5X12XIN,0X,1X12XAX,74H-4YRIN,7X,
31     *4YRAX,0X,7HPLOTTER,0(1X,F10.3)/2X,4YUCR,0(1X,F10.3))
32     105 FORMAT (47H PLOTTER AND USER AREAS DEFINED AND ARE DRAWN)
33     106 FORMAT (30H MEASURED DATA SURFACE DRAWN)
34     107 FORMAT (59H ENTER ORDER OF POLYNOMIAL IN X1- AND X2-DIRECTION, IP
35     *PRINT AND ISTAT)
36     108 FORMAT (24H SURFACE FIT COMPLETED)
37     109 FORMAT (29H APPROXIMATED SURFACE DRAWN)
38     001 FORMAT (14H/14H)
39     002 FORMAT (14H/14H)
40     003 FORMAT (14H/14H)
41     004 FORMAT (21H)
42     005 FORMAT (3F15.6)
43     C FORMATS DEMO3 STOP
44
45     : .....
46     : I/O ASSIGNMENTS.
47     :
48     : .....
49     LI=LOGUT(15554)
50     LP=0
51     LP=13
52     WRITE (LI, 101)
53     WRITE (LP, 001)
54
55     : .....
56     : INITIALIZE THE PLOTTER.
57     :
58     : .....
59     IF (.IPLOT) CALL INITS (LP)
60     WRITE (LI, 102)
61
62     : .....
63     : VARIATION OF POLYNOMIAL DEGREE. LOOP STARTS HERE.
64     :
65     : .....
66     M1STRT=1
67     M1STOP=6
68     M2STRT=1
69     M2STOP=6
70     DO 09 MORDR1=M1STRT,M1STOP,1
71     DO 09 MORDR2=M2STRT,M2STOP,1
72
73     : .....
74
75     C
76     : .....

```

```

C
C
C   READ MEASURED DATA FROM TAPE A AND SIMULTANEOUSLY SEARCH THE
C   MINIMUM AND MAXIMUM VALUE OF EACH X (X2 AND Y REQUIRED TO)
C   GET THE PLOT WITHIN THE DIMENSIONS OF A SHEET OF PAPER.
C
90  C
C
C   .....
C   READ (LD, 801) IVAL, IRAW
C   IVAL = 0.00000
C   I1MIN = 0.00000
C   I1MAX = 0.00000
C   I2MIN = 0.00000
C   I2MAX = 0.00000
C   YMIN = 0.00000
C   YMAX = 0.00000
C   IVAL = 100
C1  READ (LD, 801) IVAL, IRAW
C   IF (IVAL.EQ.101) GO TO 01
C   DECODE (30,802,IRAW) NPNTS1, NPNTS2
C   DO 03 I1=1, NPNTS1
C   DO 03 I2=1, NPNTS2
C   IVAL = 100
C2  READ (LD, 801) IVAL, IRAW
C   IF (IVAL.EQ.101) GO TO 02
C   DECODE (40,803,IRAW) X1(I1,I2), X2(I1,I2), Y1(I1,I2)
C   IF (X1(I1,I2).LT.X1MIN) X1MIN=X1(I1,I2)
C   IF (X2(I1,I2).LT.X2MIN) X2MIN=X2(I1,I2)
C   IF (Y1(I1,I2).LT.YMIN) YMIN=Y1(I1,I2)
C   IF (X1(I1,I2).GT.X1MAX) X1MAX=X1(I1,I2)
C   IF (X2(I1,I2).GT.X2MAX) X2MAX=X2(I1,I2)
C   IF (Y1(I1,I2).GT.YMAX) YMAX=Y1(I1,I2)
C3  CONTINUE
C   WRITE (LI, 103)
C
C
C   .....
C   DEFINE PLOTTER AND USER AREAS: DRAW AXES.
C   .....
C4  X0 = 10.0
C   Y0 = 10.0
C   X1PMIN = 10.0
C   X1PMAX = 10.0
C   X2PMIN = 10.0
C   X2PMAX = 10.0
C   YPMIN = 10.0
C   YPMAX = 10.0
C   X1UMIN = 10.0
C   X1UMAX = 10.0
C   X2UMIN = 10.0
C   X2UMAX = 10.0
C   YUMIN = 10.0
C   YUMAX = 10.0
C   ALFAX1 = 10.0
C   ALFAX2 = 10.0
C   ALFAY = 10.0
C   PLOT(61) = (ALFAX1*PI)/180.
C   PLOT(62) = (ALFAX2*PI)/180.
C   PLOT(63) = (ALFAY*PI)/180.
C   PLOT(64) = 0.0
C   PLOT(65) = Y0
C   PLOT(66) = (X1PMAX - X1PMIN) / (X1UMAX - X1UMIN)
C   PLOT(67) = (X2PMAX - X2PMIN) / (X2UMAX - X2UMIN)
C   PLOT(68) = (YPMAX - YPMIN) / (YUMAX - YUMIN)
C   PLOT(69) = (X1PMAX - X1PMIN) / (X1UMAX - X1UMIN)
C   PLOT(70) = (X2PMAX - X2PMIN) / (X2UMAX - X2UMIN)
C   PLOT(71) = (YPMAX - YPMIN) / (YUMAX - YUMIN)
C   X1L = (X1PMAX - X1PMIN) / (X1UMAX - X1UMIN)
C   X2L = (X2PMAX - X2PMIN) / (X2UMAX - X2UMIN)
C   YL = (YPMAX - YPMIN) / (YUMAX - YUMIN)
C   WRITE (LI, 104) X1PMIN, X1PMAX, X2PMIN, X2PMAX, YPMIN, YPMAX, X1UMIN, X1UMAX, X2UMIN, X2UMAX, YUMIN, YUMAX
C   IF (IPLOT) CALL AXES (X0,Y0,X1L,ALFAX1,2*PI*2,X1UMIN,X1UMAX,4*PI*2)
C   IF (IPLOT) CALL AXES (X0,Y0,X2L,ALFAX2,2*PI*2,X2UMIN,X2UMAX,4*PI*2)
C   IF (IPLOT) CALL AXES (X0,Y0,YL,ALFAY,2*PI*2,YUMIN,YUMAX,4*PI*2)
C   WRITE (LI, 105)
C
C
C   .....
C   DRAW MEASUREMENT DATA SURFACE.
C   .....
C
170 C

```


SUBROUTINE MAT3 (NPNTS1,NPNTS2,L,COEF,IPRINT,ISTAT,SUBM1,ERR)

PRESET SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B FOR A
3D-POLYNOMIAL APPROXIMATION. THE LINEAR EQUATION SYSTEM THEN
IS SOLVED BY USING A GAUSS JORDAN ELIMINATION.

AUTHOR: HANS ZECHNER
DATE: DECEMBER 31, 1980

VARIABLES	EXPLANATION	TYPE
NPNTS1	NUMBER OF DATA POINTS AT CONSTANT X1	INTEGER
NPNTS2	NUMBER OF DATA POINTS AT CONSTANT X2	INTEGER
COEF(49)	POLYNOMIAL COEFFICIENTS	REAL

VARIABLES	EXPLANATION	TYPE
L	NUMBER OF COEFFICIENTS FOR THE POLY-	INTEGER
M	NOMIAL IN X1 DIRECTION.	
	NUMBER OF COEFFICIENTS FOR THE POLY-	INTEGER
	NOMIAL IN X2 DIRECTION.	

VARIABLES	EXPLANATION	TYPE
IPRINT	CONTROL QUANTITY OF PRINT OUT	INTEGER
	*2 PRINT EQUATION SYSTEM BEFORE	
	AND AFTER THE GAUSS JORDAN	
	ELIMINATION.	

VARIABLES	EXPLANATION	TYPE
	*1 PRINT EQUATION SYSTEM AFTER	
	THE GAUSS JORDAN ELIMINATION.	
	*0 PRINT BASIC EQUATIONS WITH	
	ACTUAL PARAMETERS.	

VARIABLES	EXPLANATION	TYPE
ISTAT	CONTROL STATISTICS TO BE PRINTED	INTEGER
	/ISTAT/=2 PRINT ORIGINATES OF DATA	
	AND APPROXIMATED OF POINTS	
	AND DEVIATIONS (%CHANG).	

VARIABLES	EXPLANATION	TYPE
	/ISTAT/=1 ONLY PRINT ERRORS.	
	ISTAT/=2 GET ABSOLUTE ERRORS.	
	ISTAT/=3 GET RELATIVE ERRORS.	

VARIABLES	EXPLANATION	TYPE
SUBM1(M,M)	SUBMATRIX 1. CLASS IN THE SYSTEM	REAL

VARIABLES	EXPLANATION	TYPE
ERROR	ERROR FLAG RETURNED IN THE CALLING	INTEGER
	PROGRAM.	
	*1 NO ERROR DETECTED, GOOD JOB.	
	*2 AN ERROR OCCURRED IN THE GAUSSIAN	
	ELIMINATION. DEFECTIVE POINTS ARE NOT	
	ACCURATE. REVIEW PARAMETERS.	

VARIABLES	EXPLANATION	TYPE
X1(16,16)	X1-COORDINATES OF THE DATA POINTS	REAL
X2(16,16)	X2-COORDINATES OF THE DATA POINTS	REAL
Y(16,16)	Y-COORDINATES OF THE DATA POINTS	REAL
A(49,49)	SYSTEM MATRIX.	REAL
B(49)	RIGHT HAND SIDE VECTOR.	REAL

IMPORTANT: THE DATA POINTS X1, X2 AND Y HAVE TO BE PROVIDED
THROUGH THE LABELED COMMON BLOCK DATA3 AND THE
SYSTEM MATRIX A AND THE RIGHT HAND SIDE VECTOR B
ARE AVAILABLE THROUGH THE LABELED COMMON BLOCK
A03. THIS SAVES A LOT OF THAT PRECIOUS TIME
CALLED MEMORY.

COMMON / DATA3 / X1,X2,Y
COMMON / A03 / A,B

REAL X1(16,16),X2(16,16),Y(16,16)
REAL A(49,49),B(49)

REAL COEF(49),SUBM1(M,M),STAT(16,16)

C FORMATS MAT3 START

101 FORMAT (///7M AFTER 13,41M STEPS NO SOLUTION WITHIN THE ACCURACY
OF 1E-10.311M I QUIT./)
101 FORMAT (///7M ORDER OF APPROXIMATION POLYNOMIAL IN X1 DIRECTION,
95413/51M ORDER OF APPROXIMATION POLYNOMIAL IN X2 DIRECTION,95413
7740M NUMBER OF DATA POINTS WITH CONSTANT X1,16413/40M NUMBER OF
OF DATA POINTS WITH CONSTANT X2,15413/1
602 FORMAT (///62M SYSTEM MATRIX A AND VECTOR B BEFORE GAUSS JORDAN E
ELIMINATION/)
603 FORMAT (14X,13(18X,12))
604 FORMAT (24X,12(13(1X,1PE9,2),1X/(14X,13(1X,1PE9,2))))
605 FORMAT (///48M EQUATION SYSTEM AFTER GAUSS JORDAN ELIMINATION/)
606 FORMAT (///24M COEFFICIENTS COEF(1,1)/)
607 FORMAT (12X,3HJ *12,11(18X,12),1X/317X,12(18X,12)/)
608 FORMAT (21X,3HJ *12,12(14,69,31),1X/317X,12(14,69,31)/)
609 FORMAT (///30M COORDINATES OF DATA POINTS - AS GIVEN/)
610 FORMAT (///30M ABSOLUTE ERROR AT EACH POINT/)
611 FORMAT (///30M APPROXIMATED/)


```

612 FORMAT (///35H RELATIVE ERROR AT EACH POINT IN 1/)
613 FORMAT (///24H STATISTICS SUMMARY//7Y, 1H AVERAGE, 10H1, 1H 1PE4,
  2//7H, 10H MAXIMUM ERROR AT 1, 10, 100, 2//4H, 1H 1X, 1H 1PE+2)
90 614 FORMAT (///24H STATISTICS SUMMARY//7Y, 1H AVERAGE, 10H1, 1H 1PE4,
  2//7H, 10H MAXIMUM ERROR AT 1, 10, 100, 2//4H, 1H 1X, 1H 1PE+2)
615 FORMAT (///7H AFTER 1, 10, 100 STEPS NO SOLUTION WITHIN THE ACCURACY
  10H1, 1H 1PE+2)
C  FORMATS MAT3 STOP

99  LI=LOGLCISESSM1
  LO=0
  IERROR=1
  ITRMAX=50
  EPSLON=.100E-007

100  PRINT APPROXIMATION PARAMETERS, IF IPRINT IS GREATER THAN 0.
  C  C
  NOROR1=L-1
  NOROR2=M-1
105  IF (IPRINT.LT.0) GO TO 01
  WRITE (LO, 661) NOROR1, NOROR2, NPNTS1, NPNTS2
  C  C
  START TO PRESET EDGE SECTION SUBMATRICES 1. CLASS.
110  01 IPOWR1=0
  LI=2*L-1
  DO 03 K=L, LI, 1
  IF (K.GT.L) GO TO 02
115  C  C
  TOP EDGE SECTION SUBMATRICES 1. CLASS.
  ISTART=1
  ISTOP=M
  JSTART=(K-1)*M+1
  JSTOP=K*M
  GO TO 03
120  C  C
  RIGHT HAND EDGE SECTION SUBMATRICES 1. CLASS.
125  02 ISTART=(K-1)*M+1
  ISTOP=(K-1)*M+1
  JSTART=(L-1)*M+1
  JSTOP=L*M
  C  C
130  PRESET EDGE SECTION SUBMATRICES 1. CLASS IN SYSTEM MATRIX USING MAT31.
  03 CALL MAT31 (NPNTS1, NPNTS2, M, IPOWR1, SUBM1)
  I1=0
  DO 04 I=ISTART, ISTOP, 1
  I1=I+1
135  J1=0
  DO 04 J=JSTART, JSTOP, 1
  J1=J+1
  04 A(I, J)=SUBM1(I1, J1)
140  05 IPOWR1=IPOWR1+1
  C  C
  COPY DEFINED SUBMATRICES 1. CLASS DIAGONALLY.
145  LI=L-1
  DO 07 I1=L, LI, 1
  DO 07 J1=2*L, LI
  ISTART=(I1-1)*M+1
  ISTOP=I1*M
150  JSTART=(J1-1)*M+1
  JSTOP=J1*M
  DO 06 I=ISTART, ISTOP, 1
  DO 06 J=JSTART, JSTOP, 1
  A(I, J)=A(I, J)
  06 CONTINUE
155  C  C
  PRESET RIGHT HAND SIDE VECTOR B.
  I=0
  DO 08 LI=L, LI, 1
160  IPOWR1=LI-1
  DO 08 M=L, LI, 1
  IPOWR2=M-1
  08 B(I)=SINPNTS1, NPNTS2, IPOWR1, IPOWR2, 1)
165  NI=L*M
  C  C
  PRINT SYSTEM MATRIX AND RIGHT HAND SIDE VECTOR B, IF IPRINT
  IS SET TO 2.
170  IF (IPRINT.NE.2) GO TO 10

```

```

179      WRITE (LO, 602)
180      WRITE (LO, 603) (J,J=1,N1,1)
181      DO 10 J=1,N1,1
182      09 WRITE (LO, 604) [(A(I,J),J=1,N1,1),8(I)]
183      WRITE (LO, 603) (J,J=1,N1,1)
184      IPRINT=1
185      GAUSS JORDAN ELIMINATION.
186      10 EPSLON=EPSLON*4
187      CALL IEL3 (N1,ITER,ITRMAX,EPSON)
188      IF (ITER.LT.0) GO TO 27
189      PRINT EQUATION SYSTEM, IF IPRINT IS SET TO 1.
190      IF (IPRINT.NE.1) GO TO 12
191      WRITE (LO, 605)
192      WRITE (LO, 603) (J,J=1,N1,1)
193      DO 11 J=1,N1,1
194      11 WRITE (LO, 604) [(A(I,J),J=1,N1,1),8(I)]
195      WRITE (LO, 603) (J,J=1,N1,1)
196      DEFINE COEFFICIENTS AND PRINT THEM, IF IPRINT IS GREATER THAN 0.
197      12 CONTINUE
198      DO 13 I=1,N1,1
199      13 COEF(I)=8(I)
200      IF (IPRINT.LT.0) GO TO 15
201      WRITE (LO, 606)
202      WRITE (LO, 607) (I2,I2=1,N1,1)
203      DO 14 I=1,N1,1
204      14 ICSTRT=(I-1)*4+1
205      WRITE (LO, 608) I,(COEF(IC),I2=ICSTRT,ICSTP,1)
206      WRITE (LO, 607) (I2,I2=1,N1,1)
207      STATISTICS, IF ISTAT ANYTHING ELSE BUT 0.
208      15 IF (ISTAT.EQ.0) GO TO 29
209      IF (ISTAT.LT.0) ERROR=1
210      IF (ISTAT.GT.0) ERROR=2
211      ISTAT=ABS(ISTAT)
212      COMPUTE ABSOLUTE ERROR.
213      (* (APPROXIMATED VALUE - GIVEN VALUE) )
214      DO 16 I1=1,NPNTS1,1
215      DO 16 I2=1,NPNTS2,1
216      16 STAT(I1,I2)=F3(COEF,X1(I1,I2),X2(I1,I2),MOPDR1,MOPDR2)-Y(I1,I2)
217      COMPUTE RELATIVE ERROR, IF ISTAT WAS LESS THAN ZERO, I.E. ERROR=1.
218      (* (APPROXIMATED VALUE - GIVEN VALUE)/(GIVEN VALUE) ) * ERROR=1.
219      IF (ERROR.NE.1) GO TO 18
220      DO 17 I1=1,NPNTS1,1
221      DO 17 I2=1,NPNTS2,1
222      17 STAT(I1,I2)=STAT(I1,I2)/Y(I1,I2)
223      ERROR STATISTICS: COMPUTE AVERAGE ERROR AND FIND LOCATION
224      OF MAXIMUM ERROR.
225      18 ERRAVE=0
226      ERRMAX=ABS(STAT(1,1))
227      IMAZ=1
228      JMAZ=1
229      DO 19 I1=1,NPNTS1,1
230      DO 19 I2=1,NPNTS2,1
231      19 IF (ABS(STAT(I1,I2)).LE.ERRMAX) GO TO 19
232      ERRMAX=STAT(I1,I2)
233      IMAZ=I1
234      JMAZ=I2
235      ERRAVE=ERRAVE+ABS(STAT(I1,I2))
236      ERRAVE=ERRAVE/(NPNTS1*NPNTS2)
237      IF (ISTAT.NE.2) GO TO 22
238      PRINT COORDINATES OF DATA POINTS AND APPROXIMATED VALUES,
239      IF /ISTAT/ IS EQUAL 2.
240      WRITE (LO, 609)
241      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
242      DO 20 I1=1,NPNTS1,1
243      20 WRITE (LO, 604) I1,(Y(I1,I2),I2=1,NPNTS2,1)
244      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
245      WRITE (LO, 610)
246      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)

```

```

      DO 21 I1=1,NPNTS1,1
21  WRITE (LO, 604) (I1,(F3(COEF,X1(I1,I2),X2(I2,I2),NORORI,NOROK2),I2=
      *1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      ISTAT=1
260  PRINT ERRORS, IF /ISTAT/ WAS 1 OR 2.
      IF (ISTAT.NE.1) GO TO 20
265  IF (IERROR.EQ.1) GO TO 24
      PRINT ABSOLUTE ERRORS.
270  WRITE (LO, 611)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      DO 23 I1=1,NPNTS1,1
23  WRITE (LO, 604) (I1,(STAT(I1,I2),I2=1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      GO TO 20
275  PRINT RELATIVE ERRORS.
280  WRITE (LO, 612)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      DO 25 I1=1,NPNTS1,1
25  WRITE (LO, 604) (I1,(STAT(I1,I2),I2=1,NPNTS2,1)
      WRITE (LO, 603) (I2,I2=1,NPNTS2,1)
      PRINT ERROR STATISTICS.
285  IF (IERROR.EQ.2) WRITE (LO, 613) ERRAVE,IMAX,JMAX,ERRMAX
      IF (IERROR.EQ.1) WRITE (LO, 614) ERRAVE,IMAX,JMAX,ERRMAX
      GO TO 29
290  FATAL ERROR DETECTED. OUTPUT ERROR MESSAGE AND PRESET COEFFICIENTS.
27  ITER=ITER
      IERROR=IERROR
      Z(I1)=Y(I1)
295  DO 28 I2=2,N1,1
28  B(I2)=0
      WRITE (LI, 101) ITER,EPSLEN
      WRITE (LO, 615) ITER,EPSLEN
      GO TO 12
300  RETURN.
29 RETURN
      END

```

```

1      SUBROUTINE MAT31 (NPNTS1,NPNTS2,M,IPOWR1,SUBM1)
2      .....
3      * PRESET SUBMATRIX 1, CLASS SUBM1 IN THE SYSTEM MATRIX A FOR A
4      * 3D-POLYNOMIAL APPROXIMATION.
5      *
6      * AUTHOR: JANS ZERNER
7      * DATE: DECEMBER 31, 1980
8      *
9      * VARIABLE      EXPLANATION                                TYPE
10     *-----*-----*-----*-----*-----*-----*
11     * NPNTS1        NUMBER OF DATA POINTS AT CONSTANT X1      INTEGER
12     * NPNTS2        NUMBER OF DATA POINTS AT CONSTANT X2      INTEGER
13     * M             NUMBER OF COEFFICIENTS FOR THE POLY-        INTEGER
14     *               NOMIAL IN X2 DIRECTION
15     * IPOWR1        POWER FOR THE X1 VALUES IN THE             INTEGER
16     *               SUMMATION.
17     * SUBM1(M,M)    SUBMATRIX 1, CLASS IN THE SYSTEM            REAL
18     *-----*-----*-----*-----*-----*-----*
19     REAL SUBM1(M,M)
20     IPOWR2=0
21     M1=2*M-1
22     DO 03 K=1,M1,1
23     IF (K.GT.M) GO TO 01
24     TOP EDGE SECTION ELEMENTS.
25
26     I=1
27     J=4
28     GO TO 02
29     RIGHT HAND EDGE SECTION ELEMENTS.
30
31     01 I=(K-M)+1
32     J=M
33     PRESET EDGE SECTION ELEMENTS IN SUBMATRIX 1, CLASS USING S3.
34     02 SUBM1(I,J)=S3(NPNTS1,NPNTS2,IPOWR1,IPOWR2,0)
35     03 IPOWR2=IPOWR2+1
36     COPY DEFINED ELEMENTS DIAGONALLY.
37
38     M1=M-1
39     DO 04 I=1,M1,1
40     DO 04 J=2,M,1
41     04 SUBM1(I+1,J-1)=SUBM1(I,J)
42     RETURN
43     ENO
44
45
46
47
48
49
50

```

```

1      REAL FUNCTION S3 (NPNTS1,NPNTS2,IPOWR1,IPOWR2,IY)
2      .....
3      * PERFORM SUMMATIONS TO PRESET THE ELEMENTS A(I,J) IN THE
4      * SYSTEM MATRIX A AND THE ELEMENTS B(I) IN THE RIGHT HAND
5      * SIDE VECTOR B FOR THE LINEAR EQUATION SYSTEM, THAT MUST
6      * BE SOLVED FOR A POLYNOMIAL 3D-APPROXIMATION.
7      *
8      * AUTHOR: HANS ZERNER
9      * DATE: DECEMBER 31, 1980
10     *
11     * VARIABLE      EXPLANATION      TYPE
12     * .....
13     * NPNTS1        NUMBER OF DATA POINTS AT CONSTANT X1    INTEGER
14     * NPNTS2        NUMBER OF DATA POINTS AT CONSTANT X2    INTEGER
15     * IPOWR1        POWER FOR THE X1 VALUES IN THE
16     *                SUMMATION.
17     * IPOWR2        POWER FOR THE X2 VALUES IN THE
18     *                SUMMATION.
19     * IY            CONTROL PARAMETER
20     *                00 Y VALUES ARE NOT INCLUDED IN THE
21     *                SUMMATION.
22     *                01 Y VALUES ARE INCLUDED IN THE
23     *                SUMMATION.
24     *                02 Y VALUES ARE INCLUDED IN THE
25     *                SUMMATION.
26     *                03 Y VALUES ARE INCLUDED IN THE
27     *                SUMMATION.
28     *                04 Y VALUES ARE INCLUDED IN THE
29     *                SUMMATION.
30     * S3            RESULT OF SUMMATION.
31     *
32     * X1(16,16)     X1-COORDINATES OF THE DATA POINTS
33     * X2(16,16)     X2-COORDINATES OF THE DATA POINTS
34     * Y(16,16)      Y-COORDINATES OF THE DATA POINTS
35     *
36     * IMPORTANT: THE DATA POINTS X1, X2 AND Y HAVE TO BE PROVIDED
37     * THROUGH THE LABELED COMMON BLOCK DATA3.
38     *
39     * .....
40     COMMON / DATA3 / X1,X2,Y
41     REAL X1(16,16),X2(16,16),Y(16,16)
42     S=0
43     DO 03 K1=1,NPNTS1
44     DO 03 K2=1,NPNTS2
45     IF (IPOWR1.EQ.0) GO TO 01
46     S1=X1(K1,K2)**IPOWR1
47     GO TO 02
48     IF (IPOWR2.EQ.0) GO TO 03
49     S2=X2(K1,K2)**IPOWR2
50     GO TO 04
51     S2=1.
52     IF (IY.EQ.0) SY=1.
53     S=S1*S2*SY
54     RETURN
55     END

```

1 INTEGER FUNCTION IEL33 (N)

5
 6 GAUSS JORDAN ELIMINATION WITH MATRIX A AND VECTOR B. THE
 7 VARIABLE IEL3 IS SET TO 1 IF THE ELIMINATION IS COMPLETED.
 8 IEL3 LESS THAN 0 INDICATES AN ERROR.

9
 10 AUTHOR: HANS ZEBNER
 11 DATE: DECEMBER 31, 1980

12
 13 VARIABLE EXPLANATION TYPE
 14 N NUMBER OF EQUATIONS INTEGER
 15 IEL3 CONTROL PARAMETER, RETURNED TO THE CALLING ROUTINE. INTEGER
 16 01 GAUSS JORDAN ELIMINATION WAS PERFORMED CORRECTLY.
 17 00 ERROR ENCOUNTERED DURING GAUSS JORDAN ELIMINATION.

18
 19 A(49,49) SYSTEM MATRIX A REAL
 20 B(49) RIGHT HAND SIDE VECTOR B REAL

21
 22 IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH THE LABELED COMMON BLOCK A83. BOTH A AND B WILL UNDERGO CHANGES DURING EXECUTION, I.E. THE ELIMINATION.

30
 31 COMMON / A83 / A,B

32 REAL A(49,49),B(49)

35 IEL3=1
 36 IEL33=IEL3
 37 EPS=1.00E-030
 38 DO 06 I=1,N+1
 39 DO 02 K=1,N+1
 40 IF (K.EQ.I) GO TO 02
 41 CONST=A(K,I)/A(I,I)
 42 DO 01 J=1,N+1
 43 A(K,J)=A(K,J)+CONST*A(I,J)
 44 IF (J.EQ.I) A(K,J)=0
 45 01 CONTINUE
 46 B(K)=B(K)+CONST*B(I)
 47 02 CONTINUE
 48 CONST=A(I,I)
 49 DO 03 J=1,N+1
 50 A(I,J)=A(I,J)/CONST
 51 A(I,I)=1
 52 B(I)=B(I)/CONST
 53 DO 06 J=1,N+1
 54 ERROR=0
 55 DO 04 K=1,N+1
 56 IF (ABS(A(I,K))-GT.EPS) GO TO 34
 57 ERROR=ERROR+1
 58 04 CONTINUE
 59 IF (ABS(B(I))-GT.EPS) GO TO 35
 60 ERROR=ERROR+1
 61 IF (ERROR.GT.N+1) GO TO 37
 62 06 CONTINUE
 63 GO TO 08
 64 IEL3=-1
 65 IEL33=IEL3
 66 RETURN
 67 END

```

1      SUBROUTINE IEL3 (N,ITER,ITRMAX,EPSLON)
2      .....
3      * GAUSSIAN ELIMINATION WITH LINEAR EQUATION SYSTEM V (THE RIGHT
4      * HAND SIDE VECTOR) IS INTEGRATED TOGETHER WITH THE SYSTEM *A*
5      * *A* IN THE MATRIX V). ITER IS THE NUMBER OF ITERATIONS *A*
6      * TURNED TO THE CALLING PROGRAM. THE ITERATION TO IMPROVE THE
7      * SOLUTION SO MANY TIMES. UNTIL THE REMAINING ERROR IS WITHIN
8      * THE MAGNITUDE OF EPSLON. SHOULD THE NUMBER OF ITERATIONS EX-
9      * CEED ITRMAX, THE SUBROUTINE TERMINATES THE CALCULATION AND
10     * SETS ITER TO A NEGATIVE VALUE.
11     *
12     * AUTHORS: KONRAD VOGELER, HANS ZEBHER
13     * DATE: DECEMBER 31, 1980
14     *
15     * VARIABLE EXPLANATION TYPE
16     * .....
17     * N ..... NUMBER OF EQUATIONS ..... INTEGER
18     * ITER ..... NUMBER OF ITERATIONS TO BE RETURNED ..... INTEGER
19     * ..... NO GAUSSIAN ELIMINATION PERFORMED .....
20     * ..... CORRECTLY. ....
21     * ..... NO CONVERGENCE OF THE SOLUTION FOR
22     * ..... THE DESIRED RESOLUTION EPSLON.
23     * ..... ITERATION DID TERMINATE AFTER
24     * ..... ITRMAX ITERATION STEPS.
25     * ITRMAX ..... MAXIMUM NUMBER OF ITERATION STEPS ..... INTEGER
26     * EPSLON ..... DESIRED RESOLUTION OF THE SOLUTION ..... REAL
27     *
28     * A(49:49) SYSTEM MATRIX A ..... 49x49
29     * B(49) RIGHT HAND SIDE VECTOR B ..... REAL
30     *
31     * IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH
32     * THE LABELED COMMON BLOCK A33. BOTH A AND B WILL
33     * STILL UNDERGO CHANGES DURING EXECUTION, I.E. THE
34     * ELIMINATION.
35     *
36     * .....
37     * COMMON / A33 / V
38     * REAL V(49,50),G(50),A(49,50),B(50)
39     * LOGICAL IPRINT
40     * DATA IPRINT / .FALSE. /
41     * DATA PI / 3.141592654 /
42     *
43     * FORMATS IEL3 START
44     * 601 FORMAT (///34H MATRIX A AND VECTOR B VOR ELIMINATION/)
45     * 602 FORMAT ((4X,13(8X,12)))
46     * 603 FORMAT (2X,12,13(1X,12),21(1X,13(1X,12),12),12)
47     * 604 FORMAT (///25H REVOE SYSTEM (M=23,14, ITERATIONSSCHRITT/
48     * )
49     * 605 FORMAT (///2X,12,24H Koeffizienten aus Gauss/)
50     * 606 FORMAT ((2X,12,1X,12),12)
51     * 607 FORMAT (///34H Koeffizienten Zellenvektorschnittwinkel *e12,3)
52     * 608 FORMAT (2X,12,4 RESIDUUM *e12,3,5H (A33,13,12, ITERATIONEN)
53     * 609 FORMAT (///20X,11,12,3,5H (A33,13,12, ITERATIONEN)
54     * 610 FORMAT (///25H SYSTEM NACH ELIMINATION/)
55     *
56     * FORMATS IEL3 STOP
57     *
58     * LG=6
59     * NM=1
60     * KM=1
61     *
62     * REARRANGE VECTOR B IN ARRAY V, IF N IS LESS THAN 49.
63     * IF (N.EQ.49) GO TO 32
64     * DO 01 I=1,N,1
65     * V(I,N)=V(I,50)
66     * 01
67     * PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B BEFORE THE
68     * ELIMINATION, IF IPRINT IS SET .TRUE.
69     *
70     * 02 IF (IPRINT) WRITE (LG, 601)
71     * IF (IPRINT) WRITE (LG, 602) ((J,J=1,N,1)
72     * DO 03 I=1,N,1
73     * IF (IPRINT) WRITE (LG, 603) (V(I,J),J=1,N,1)
74     * IF (IPRINT) WRITE (LG, 602) ((J,J=1,N,1)
75     *
76     * LINEAR EQUATION SYSTEM V(N,N)*G(N)=V(N,M+1)
77     *
78     * DO 04 I=1,M,1
79     * DO 04 J=1,M,1
80     * A(I,J)=V(I,J)
81     * DO 05 I=1,M,1
82     * 05 B(I)=0
83     *
84     * CHECK FOR STABILITY.

```

C

```

      FI=.0
      DO 06 I=1,M,1
06      G(I)=.0
      DO 08 J=1,N,1
      DO 07 I=1,M,1
07      G(I)=G(I)+V(I,J)**2
08      G(I)=SQRT(G(I))
      DO 10 J=1,N,1
      DO 10 J=J,N,1
      SK=.0
      DO 09 I=1,M,1
09      SK=SK+V(I,J)*V(I,J)
      F=SK/G(I)*G(J)
      IF (ABS(F)-.01*ABS(FI)) FI=F
10      CONTINUE
      F=ABS(FI)
      F=ACOS(FI)
      F=(F+180.)/PI

```

START ITERATION TO FIND AND IMPROVE SOLUTION.

```

      DO 27 LX=1,999,1
      ITER=LX
      DO 15 J=1,N,1

```

PIVOTISATION AND GAUSSIAN STEP FOR COLUMN J.

```

      IX=J
      IX=J
      DO 12 I=J,N,1
      IF (ABS(V(I,J))-GT.ABS(E)) GO TO 11
      GO TO 12
11      IX=I
      E=V(I,J)
12      CONTINUE
      IF (IX.EQ.J) GO TO 14
      DO 13 I=1,M,1
      G(I)=V(I,IX)
      V(I,IX)=V(I,J)
13      V(I,J)=G(I)
      G(I)=.0
14      MM=J+1
      DO 15 I=MM,N,1
      DO 15 K=J,M
      V(I,KK)=V(I,KK)-V(I,MM)*V(I,KK)/E
15      CONTINUE
      IF (V(I,I).EQ..0) GO TO 32

```

SOLUTION.

```

      DO 16 I=1,M,1
16      G(I)=.0
      G(M)=V(M,M)/V(I,M)
      DO 19 II=1,K,1
      I=M-II
      IF (V(I,I).EQ..0) GO TO 32
      G(II)=V(I,M)
      DO 17 JJ=1,K,1
      JJ=J
      IF (JJ.EQ.I) GO TO 19
17      G(II)=G(II)-G(JJ)*V(I,JJ)
19      CONTINUE
      G(II)=G(II)/V(I,I)

```

CORRECT SOLUTION BY AMOUNT OF ERROR, WHICH WAS INVOLVED IN THIS ITERATION STEP. TAKE THIS VECTOR AS START VECTOR FOR THE NEXT ITERATION.

```

      DO 20 I=1,M,1
20      G(II)=G(II)+G(I)

```

COMPUTE THE REMAINING ERROR.

```

      DO 22 I=1,M,1
      V(I,M)=.0
      DO 21 J=1,N,1
21      V(I,M)=V(I,M)+G(J)*A(I,J)
22      V(I,M)=A(I,M)-V(I,M)
      R=.0
      DO 23 I=1,M,1
23      R=R+V(I,M)*V(I,M)

```



```

      R=SQRT(R)
      IF (R.LT.EPSLOH) GO TO 28
      IF (L.LT.(TRMAX)) GO TO 33
      DO 24 I=1,N+1
175      24 B(I)=G(I)
      PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B AFTER THE
      PIVOTISATION.
180      IF ((PRINT) WRITE (LJ, 604) LX
      IF ((PRINT) WRITE (LJ, 602) (J,J=1,N+1)
      DO 25 I=1,N+1
185      25 IF ((PRINT) WRITE (LJ, 603) (,V(I,J),J=1,N+1),G(I)
      IF ((PRINT) WRITE (LJ, 602) (J,J=1,N+1)
      DO 26 I=1,N+1
      DO 26 J=1,N+1
      26 V(I,J)=A(I,J)
      STOP LOOP TO FIND SOLUTION.
190      27 CONTINUE
      REARRANGE SOLUTION IN THE ARRAY V.
195      28 CONTINUE
      DO 29 I=1,N+1
      29 V(I,50)=G(I)
      PRINT SOLUTION, IF IPRINT IS SET .TRUE. .
200      30 IF ((PRINT) WRITE (LJ, 605) N
      IF ((PRINT) WRITE (LJ, 606) (,G(I),I=1,N+1)
      IF ((PRINT) WRITE (LJ, 607) FL
      IF ((PRINT) WRITE (LJ, 608) R,LX
205      DRUCKEN DER MATRIX A UND DES VECTORS B NACH ELIMINATION.
      IF ((PRINT) WRITE (LJ, 610)
      IF ((PRINT) WRITE (LJ, 602) (J,J=1,N+1)
      DO 31 I=1,N+1
210      31 IF ((PRINT) WRITE (LJ, 603) (,V(I,J),J=1,N+1)
      IF ((PRINT) WRITE (LJ, 602) (J,J=1,N+1)
      RETURN
215      ERROR ENCOUNTERED. STOP.
      32 ITER=1
      WRITE (LJ, 609) I,I,V(I,I)
      GO TO 30
      RETURN
220      NO CONVERGENCE OF THE SOLUTION.
      33 ITER=ITER
      RETURN
225      END

```

FUNCTION F3

14/175 DPT=2 P10MP

FTN 4.00443

00/12/29 14:54.9

0302

```

1      REAL FUNCTION F3 (COEF,X1,X2,NORDR1,NORDR2)
2      .....
3      * COMPUTE FUNCTION VALUE Y=F(X1,X2) OF A 3D-POLYNOMIAL USING
4      * THE HORNER SCHEME IN ORDER TO INCREASE SPEED AND ACCURACY.
5      *
6      * AUTHOR: HANS ZEHNER
7      * DATE: DECEMBER 31, 1989
8      *
9      * VARIABLE EXPLANATION TYPE
10     * COEF(49) POLYNOMIAL COEFFICIENTS REAL
11     * X1 X1-VALUE, WHERE Y WILL BE COMPUTED. REAL
12     * X2 X2-VALUE, WHERE Y WILL BE COMPUTED. REAL
13     * NORDR1 ORDER OF THE POLYNOMIAL IN THE INTEGER
14     * NORDR2 X1 DIRECTION. ORDER OF THE POLYNOMIAL IN THE
15     * X2 DIRECTION. INTEGER
16     * F3 ORIGINATE OF THE 3D-POLYNOMIAL. REAL
17     *
18     *
19     *
20     *
21     *
22     *
23     *
24     *
25     *
26     *
27     *
28     *
29     *
30     *
31     *
32     *
33     *
34     *
35     *
36     *
37     *
38     *
39     *
40     *
41     *
42     *
43     *
44     *
45     *
46     *
47     *
48     *
49     *

```

BLOCK DATA AB4 74/175 JPT-2 P40MP

FIN 4.84443

50/22/29. 14.25.24

PAGE 1

1

BLOCK DATA AB4
COMMON / AB4 / A,B
REAL 1(64.04),B(04)
END

BLOCK DATA DATA4

74/175 JPT=2 PMONP

FTM +.0+4.75

SC/12/29. 14.12.13

PAGE

1

1

BLOCK DATA DATA4

COMMON / DATA4 / X1,X2,X3,Y

REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)

END

```

1      PROGRAM DEMO4 (INPUT,OUTPUT,TAPE1,TAPE6,TAPE7,TAPE3=INPUT)
2      .....
3      : THIS IS A DEMONSTRATION PROGRAM AND IT SHOWS THE CORRECT USE
4      : OF THE TPL BINARY LIBRARY USING EXAMPLES.
5      :
6      : AUTHOR:      HANS ZERNER
7      : DATE:        DECEMBER 31, 1980
8      :
9      .....
10     COMMON / AFLO / PLOTR
11     COMMON / DATA4 / X1,X2,X3,Y
12
13     REAL PLOTR(256)
14     REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)
15
16     REAL COEF(64),SUBM1(49,49),SUBM2(7,7)
17     INTEGER IRAN(6)
18     LOGICAL IPLOT
19
20     DATA ICMT / 1000 /
21     DATA IPLOT / .TRUE. /
22     DATA PI / 3.141592654 /
23
24     C FORMATS DEMO4 START
25     101 FORMAT (//72H ***** THIS IS PROGRAM DEMO4 AND WE'LL SEE, HOW I
26     *O USE THE MARVELLOUS/70M TPL BINARY LIBRARY. YOUR INTEREST WILL
27     *BE GREATLY AWAKED BY EASIER/25H AND FASTER PROGRAMMING-//)
28
29     102 FORMAT (22H PLOTTER INITIALIZED.)
30     103 FORMAT (31H DATA POINTS READ FROM TAPE 7.)
31     104 FORMAT (49H 5411MIN-48.5411MAX-08.5412MIN-08.5412MAX-78.447MIN-78.
32     *447MAX-08.744PLTTER-01-X-PL-31/98-ANUCP-01(X,PIC,3))
33     105 FORMAT (47H PLOTTER AND USER ARE DEFINED AND AXES DRAWN.)
34     106 FORMAT (31H MEASURED DATA SURFACES DRAWN.)
35     107 FORMAT (74H ENTER ORDER OF POLYNOMIAL IN X1-, X2- AND X3-DIRECTIO
36     *N, IPRINT AND ISTAT)
37     108 FORMAT (25H SURFACE FITS COMPLETED.)
38     109 FORMAT (30H APPROXIMATED SURFACES DRAWN.)
39     110 FORMAT (14H/14H)
40     111 FORMAT (A1,8A13)
41     112 FORMAT (31F1)
42     113 FORMAT (4F15.6)
43     C FORMATS DEMO4 STOP
44
45     C ***** I/O ASSIGNMENTS. *****
46     :
47     :
48     :
49     :
50     LI=LOGUTISSESHI
51     LO=7
52     LP=13
53     WRITE (LI, 101)
54     WRITE (LO, 601)
55
56     C ***** INITIALIZE THE PLOTTER. *****
57     :
58     :
59     :
60     :
61     IF (IPLOT) CALL INITG (LP)
62     WRITE (LI, 102)
63
64     C ***** VARIATION OF POLYNOMIAL DEGREE. LOOP STARTS HERE. *****
65     :
66     :
67     :
68     :
69     M1STRT=1
70     M1STOP=6
71     M2STRT=1
72     M2STOP=6
73     M3STRT=1
74     M3STOP=6
75     DO 10 MORO1=M1STRT,M1STOP,1
76     DO 10 MORO2=M2STRT,M2STOP,1
77     DO 10 MORO3=M3STRT,M3STOP,1
78     ITH=(MORO1+1)*(MORO2+1)*(MORO3+1)
79     IF (ITH.GT.64) GO TO 10
80
81     C *****
82     :
83     :
84     :
85     :
86     :
87     :
88     :
89     :
90     :
91     :
92     :
93     :
94     :
95     :
96     :
97     :
98     :
99     :
100    :
101    :
102    :
103    :
104    :
105    :
106    :
107    :
108    :
109    :
110    :
111    :
112    :
113    :
114    :
115    :
116    :
117    :
118    :
119    :
120    :
121    :
122    :
123    :
124    :
125    :
126    :
127    :
128    :
129    :
130    :
131    :
132    :
133    :
134    :
135    :
136    :
137    :
138    :
139    :
140    :
141    :
142    :
143    :
144    :
145    :
146    :
147    :
148    :
149    :
150    :
151    :
152    :
153    :
154    :
155    :
156    :
157    :
158    :
159    :
160    :
161    :
162    :
163    :
164    :
165    :
166    :
167    :
168    :
169    :
170    :
171    :
172    :
173    :
174    :
175    :
176    :
177    :
178    :
179    :
180    :
181    :
182    :
183    :
184    :
185    :
186    :
187    :
188    :
189    :
190    :
191    :
192    :
193    :
194    :
195    :
196    :
197    :
198    :
199    :
200    :
201    :
202    :
203    :
204    :
205    :
206    :
207    :
208    :
209    :
210    :
211    :
212    :
213    :
214    :
215    :
216    :
217    :
218    :
219    :
220    :
221    :
222    :
223    :
224    :
225    :
226    :
227    :
228    :
229    :
230    :
231    :
232    :
233    :
234    :
235    :
236    :
237    :
238    :
239    :
240    :
241    :
242    :
243    :
244    :
245    :
246    :
247    :
248    :
249    :
250    :
251    :
252    :
253    :
254    :
255    :
256    :
257    :
258    :
259    :
260    :
261    :
262    :
263    :
264    :
265    :
266    :
267    :
268    :
269    :
270    :
271    :
272    :
273    :
274    :
275    :
276    :
277    :
278    :
279    :
280    :
281    :
282    :
283    :
284    :
285    :
286    :
287    :
288    :
289    :
290    :
291    :
292    :
293    :
294    :
295    :
296    :
297    :
298    :
299    :
300    :
301    :
302    :
303    :
304    :
305    :
306    :
307    :
308    :
309    :
310    :
311    :
312    :
313    :
314    :
315    :
316    :
317    :
318    :
319    :
320    :
321    :
322    :
323    :
324    :
325    :
326    :
327    :
328    :
329    :
330    :
331    :
332    :
333    :
334    :
335    :
336    :
337    :
338    :
339    :
340    :
341    :
342    :
343    :
344    :
345    :
346    :
347    :
348    :
349    :
350    :
351    :
352    :
353    :
354    :
355    :
356    :
357    :
358    :
359    :
360    :
361    :
362    :
363    :
364    :
365    :
366    :
367    :
368    :
369    :
370    :
371    :
372    :
373    :
374    :
375    :
376    :
377    :
378    :
379    :
380    :
381    :
382    :
383    :
384    :
385    :
386    :
387    :
388    :
389    :
390    :
391    :
392    :
393    :
394    :
395    :
396    :
397    :
398    :
399    :
400    :
401    :
402    :
403    :
404    :
405    :
406    :
407    :
408    :
409    :
410    :
411    :
412    :
413    :
414    :
415    :
416    :
417    :
418    :
419    :
420    :
421    :
422    :
423    :
424    :
425    :
426    :
427    :
428    :
429    :
430    :
431    :
432    :
433    :
434    :
435    :
436    :
437    :
438    :
439    :
440    :
441    :
442    :
443    :
444    :
445    :
446    :
447    :
448    :
449    :
450    :
451    :
452    :
453    :
454    :
455    :
456    :
457    :
458    :
459    :
460    :
461    :
462    :
463    :
464    :
465    :
466    :
467    :
468    :
469    :
470    :
471    :
472    :
473    :
474    :
475    :
476    :
477    :
478    :
479    :
480    :
481    :
482    :
483    :
484    :
485    :
486    :
487    :
488    :
489    :
490    :
491    :
492    :
493    :
494    :
495    :
496    :
497    :
498    :
499    :
500    :
501    :
502    :
503    :
504    :
505    :
506    :
507    :
508    :
509    :
510    :
511    :
512    :
513    :
514    :
515    :
516    :
517    :
518    :
519    :
520    :
521    :
522    :
523    :
524    :
525    :
526    :
527    :
528    :
529    :
530    :
531    :
532    :
533    :
534    :
535    :
536    :
537    :
538    :
539    :
540    :
541    :
542    :
543    :
544    :
545    :
546    :
547    :
548    :
549    :
550    :
551    :
552    :
553    :
554    :
555    :
556    :
557    :
558    :
559    :
560    :
561    :
562    :
563    :
564    :
565    :
566    :
567    :
568    :
569    :
570    :
571    :
572    :
573    :
574    :
575    :
576    :
577    :
578    :
579    :
580    :
581    :
582    :
583    :
584    :
585    :
586    :
587    :
588    :
589    :
590    :
591    :
592    :
593    :
594    :
595    :
596    :
597    :
598    :
599    :
600    :
601    :
602    :
603    :
604    :
605    :
606    :
607    :
608    :
609    :
610    :
611    :
612    :
613    :
614    :
615    :
616    :
617    :
618    :
619    :
620    :
621    :
622    :
623    :
624    :
625    :
626    :
627    :
628    :
629    :
630    :
631    :
632    :
633    :
634    :
635    :
636    :
637    :
638    :
639    :
640    :
641    :
642    :
643    :
644    :
645    :
646    :
647    :
648    :
649    :
650    :
651    :
652    :
653    :
654    :
655    :
656    :
657    :
658    :
659    :
660    :
661    :
662    :
663    :
664    :
665    :
666    :
667    :
668    :
669    :
670    :
671    :
672    :
673    :
674    :
675    :
676    :
677    :
678    :
679    :
680    :
681    :
682    :
683    :
684    :
685    :
686    :
687    :
688    :
689    :
690    :
691    :
692    :
693    :
694    :
695    :
696    :
697    :
698    :
699    :
700    :
701    :
702    :
703    :
704    :
705    :
706    :
707    :
708    :
709    :
710    :
711    :
712    :
713    :
714    :
715    :
716    :
717    :
718    :
719    :
720    :
721    :
722    :
723    :
724    :
725    :
726    :
727    :
728    :
729    :
730    :
731    :
732    :
733    :
734    :
735    :
736    :
737    :
738    :
739    :
740    :
741    :
742    :
743    :
744    :
745    :
746    :
747    :
748    :
749    :
750    :
751    :
752    :
753    :
754    :
755    :
756    :
757    :
758    :
759    :
760    :
761    :
762    :
763    :
764    :
765    :
766    :
767    :
768    :
769    :
770    :
771    :
772    :
773    :
774    :
775    :
776    :
777    :
778    :
779    :
780    :
781    :
782    :
783    :
784    :
785    :
786    :
787    :
788    :
789    :
790    :
791    :
792    :
793    :
794    :
795    :
796    :
797    :
798    :
799    :
800    :
801    :
802    :
803    :
804    :
805    :
806    :
807    :
808    :
809    :
810    :
811    :
812    :
813    :
814    :
815    :
816    :
817    :
818    :
819    :
820    :
821    :
822    :
823    :
824    :
825    :
826    :
827    :
828    :
829    :
830    :
831    :
832    :
833    :
834    :
835    :
836    :
837    :
838    :
839    :
840    :
841    :
842    :
843    :
844    :
845    :
846    :
847    :
848    :
849    :
850    :
851    :
852    :
853    :
854    :
855    :
856    :
857    :
858    :
859    :
860    :
861    :
862    :
863    :
864    :
865    :
866    :
867    :
868    :
869    :
870    :
871    :
872    :
873    :
874    :
875    :
876    :
877    :
878    :
879    :
880    :
881    :
882    :
883    :
884    :
885    :
886    :
887    :
888    :
889    :
890    :
891    :
892    :
893    :
894    :
895    :
896    :
897    :
898    :
899    :
900    :
901    :
902    :
903    :
904    :
905    :
906    :
907    :
908    :
909    :
910    :
911    :
912    :
913    :
914    :
915    :
916    :
917    :
918    :
919    :
920    :
921    :
922    :
923    :
924    :
925    :
926    :
927    :
928    :
929    :
930    :
931    :
932    :
933    :
934    :
935    :
936    :
937    :
938    :
939    :
940    :
941    :
942    :
943    :
944    :
945    :
946    :
947    :
948    :
949    :
950    :
951    :
952    :
953    :
954    :
955    :
956    :
957    :
958    :
959    :
960    :
961    :
962    :
963    :
964    :
965    :
966    :
967    :
968    :
969    :
970    :
971    :
972    :
973    :
974    :
975    :
976    :
977    :
978    :
979    :
980    :
981    :
982    :
983    :
984    :
985    :
986    :
987    :
988    :
989    :
990    :
991    :
992    :
993    :
994    :
995    :
996    :
997    :
998    :
999    :
1000   :

```

```

90      .....
95      : READ MEASURED DATA FROM TAPEY AND SIMULTANEOUSLY SEARCH THE :
      : MINIMUM AND MAXIMUM VALUE OF EACH X1, X2 AND Y (REQUIRED TO :
      : GET THE PLOT WITHIN THE BOUNDARIES OF A SHEET OF PAPER). :
      .....
      REMIND LO
      IVAL = 10M
      X1MIN = -99999.
      X1MAX = -99999.
      X2MIN = -99999.
      X2MAX = -99999.
      YMIN = -99999.
      YMAX = -99999.
100      .....
105      01 READ (LO, 7C1) IVAL, IRAM
      IF (IVAL.EQ.1) GO TO 01
      DECODE (45,702,IRAM) NPNTS1,NPNTS2,NPNTS3
      DO 03 I3=1,NPNTS3+1
      DO 03 I1=1,NPNTS1+1
      DO 03 I2=1,NPNTS2+1
110      IVAL=10M
      02 READ (LO, 7C1) IVAL, IRAM
      IF (IVAL.EQ.1) GO TO 01
      DECODE (60,7C3,IRAM) X1(I1),X2(I1),X2(I2),X2(I3),X3(I1),X3(I2),X3(I3),Y(I1),Y(I2),Y(I3)
115      IF (X1(I1),X2(I3).LT.X1MIN) X1MIN=X1(I1),X2(I3)
      IF (X2(I1),X2(I3).LT.X2MIN) X2MIN=X2(I1),X2(I3)
      IF (Y(I1),X2(I3).LT.YMIN) YMIN=Y(I1),X2(I3)
      IF (X1(I1),X2(I3).GT.X1MAX) X1MAX=X1(I1),X2(I3)
      IF (X2(I1),X2(I3).GT.X2MAX) X2MAX=X2(I1),X2(I3)
      IF (Y(I1),X2(I3).GT.YMAX) YMAX=Y(I1),X2(I3)
120      03 CONTINUE
      WRITE (LI, 103)

125      .....
130      : DEFINE PLOTTER AND USER AREAS; DRAW AXES. :
      .....
      IO = 1
      YO = 1.
      X1PMIN = 10.0
      X1PMAX = 10.0
      X2PMIN = 10.0
      X2PMAX = 10.0
      YPMIN = 10.0
      YPMAX = 10.0
      X1UMIN = X1MIN
      X1UMAX = X1MAX
      X2UMIN = X2MIN
      X2UMAX = X2MAX
      YUMIN = YMIN
      YUMAX = YMAX
      ALFA11 = 90.0
      ALFA12 = 120.0
      ALFA1 = 180.0
      PLOT1(51) = (ALFA11*PI)/180.
      PLOT1(52) = (ALFA12*PI)/180.
      PLOT1(53) = (ALFA1*PI)/180.
      PLOT1(54) = XO
      PLOT1(55) = YO
      PLOT1(56) = (X1PMAX - X1PMIN)/(X1UMAX - X1UMIN)
      PLOT1(57) = (X2PMAX - X2PMIN)/(X2UMAX - X2UMIN)
      PLOT1(58) = (X1PMAX - X2PMIN)/(X1UMAX - X2UMIN)
      PLOT1(59) = (X2PMAX - X2UMIN)/(X2UMAX - X2UMIN)
      PLOT1(70) = (YPMAX - YPMIN)/(YUMAX - YUMIN)
      PLOT1(71) = (YPMAX - YUMIN)/(YUMAX - YUMIN)
      X1L = (X1PMAX - X1PMIN)
      X2L = (X2PMAX - X2PMIN)
      YL = (YPMAX - YPMIN)
      WRITE (LI, 104) X1MIN,X1MAX,X2MIN,X2MAX,YPMIN,YPMAX,X1UMIN,X1U
      MAX,X2UMIN,X2UMAX,YUMIN,YUMAX
      IF (IPL0T) CALL AXES (XO,YO,X1L,ALFA1,2,X1L,2,X1UMIN,X1UMAX,4,4F7,
      3,7,8)
      IF (IPL0T) CALL AXES (XO,YO,X2L,ALFA2,2,X2L,2,X2UMIN,X2UMAX,4,4F7,
      3,7,8)
      IF (IPL0T) CALL AXES (XO,YO,YL,ALFA1,3,YL,2,YUMIN,YUMAX,4,4F7,
      3,7,8)
      WRITE (LI, 105)
170

```

```

175      .....
      : DRAW MEASUREMENT DATA SURFACES.
      : .....
180      00 04 I3=L,NPNTS3,1
      00 04 I2=L,NPNTS2,1
      00 04 I1=L,NPNTS1,1
      IF (IPLOT) CALL THRTW (XPL0T,YPL0T,X1(I1,I2,I3),X2(I1,I2,I3),Y(I1,
      *I2,I3))
185      IF (IPLOT .AND. I2.EQ.1) CALL PLOT (XPL0T,YPL0T,3)
      04 IF (IPLOT .AND. I2.GT.1) CALL PLOT (XPL0T,YPL0T,2)
      00 05 I3=L,NPNTS3,1
      00 05 I2=L,NPNTS2,1
      00 05 I1=L,NPNTS1,1
      IF (IPLOT) CALL THRTW (XPL0T,YPL0T,X1(I1,I2,I3),X2(I1,I2,I3),Y(I1,
      *I2,I3))
190      IF (IPLOT .AND. I1.EQ.1) CALL PLOT (XPL0T,YPL0T,3)
      05 IF (IPLOT .AND. I1.GT.1) CALL PLOT (XPL0T,YPL0T,2)
      WRITE (LI, 100)
195
200      .....
      : CALCULATE SURFACE FITS THROUGH DATA POINTS.
      : .....
205      06 WRITE (LI, 107)
      READ (9, *) MOROR1,MOROR2,MOROR3,IPRINT,ISTAT
      MOROR1=MOROR1
      MOROR2=MOROR2
      MOROR3=MOROR3
      IPRINT=0
      ISTAT=1
      L=MOROR1+1
      M=MOROR2+1
      N=MOROR3+1
      MN=M*N
      IF (L.LT.1 .OR. L.GT.7) GO TO 06
      IF (M.LT.1 .OR. M.GT.7) GO TO 06
      IF (N.LT.1 .OR. N.GT.7) GO TO 06
      IF (L*M*N.LT.1 .OR. L*M*N.GT.84) GO TO 06
      CALL FATS (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT,ISTAT,SUBM1,MN,S
      *UBM2,ERROR)
      IF (ERROR.LT.0) GO TO 09
      WRITE (LI, 108)
220
225      .....
      : DRAW APPROXIMATED SURFACES.
      : .....
230      00 07 I3=L,NPNTS3,1
      00 07 I2=L,NPNTS2,1
      00 07 I1=L,NPNTS1,1
      YAPP=F(COEF,X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3),MOROR1,MOROR2,
      *MOROR3)
      IF (IPLOT) CALL THRTW (XPL0T,YPL0T,X1(I1,I2,I3),X2(I1,I2,I3),YAPP)
235      IF (IPLOT .AND. I2.EQ.1) CALL PLOT (XPL0T,YPL0T,3)
      07 IF (IPLOT .AND. I2.GT.1) CALL PLOT (XPL0T,YPL0T,2)
      00 08 I3=L,NPNTS3,1
      00 08 I2=L,NPNTS2,1
      00 08 I1=L,NPNTS1,1
      YAPP=F(COEF,X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3),MOROR1,MOROR2,
      *MOROR3)
      IF (IPLOT) CALL THRTW (XPL0T,YPL0T,X1(I1,I2,I3),X2(I1,I2,I3),YAPP)
      IF (IPLOT .AND. I1.EQ.1) CALL PLOT (XPL0T,YPL0T,3)
240      08 IF (IPLOT .AND. I1.GT.1) CALL PLOT (XPL0T,YPL0T,2)
      245 WRITE (LI, 109)
      09 IF (IPLOT) CALL NUPAGE
      10 CONTINUE
250
255      .....
      : TERMINATE GRAPHICS.
      : .....

```

PROGRAM DE104 747175 OPT=2 PMONP

FTM 4.000000

06/12/1964 14000000

PAGE 4

IF (IPL0T) CALL ST0PG

260

STOP 0077
END


```

1  SUBROUTINE MAT4 (NPNTS1,NPNTS2,NPNTS3,L,M,N,COEF,IPRINT,ISTAT,SUBM
2  *1,M,SUBM2,ERROR)
3  .....
4  PRESET SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B FOR A
5  40-POLYNOMIAL APPROXIMATION. THE LINEAR EQUATION SYSTEM THEN
6  IS SOLVED BY USING A GAUSS JORDAN ELIMINATION.
7  .....
8  AUTHOR: HANS ZEBNER
9  DATE: DECEMBER 31, 1980
10 .....
11 VARIABLE EXPLANATION TYPE
12 .....
13 NPNTS1 NUMBER OF DATA POINTS AT CONSTANT X1 INTEGER
14 NPNTS2 NUMBER OF DATA POINTS AT CONSTANT X2 INTEGER
15 NPNTS3 NUMBER OF DATA POINTS AT CONSTANT X3 INTEGER
16 COEF(64) POLYNOMIAL COEFFICIENTS REAL
17 L NUMBER OF COEFFICIENTS FOR THE POLY- INTEGER
18 NOMIAL IN X1 DIRECTION
19 M NUMBER OF COEFFICIENTS FOR THE POLY- INTEGER
20 NOMIAL IN X2 DIRECTION
21 N NUMBER OF COEFFICIENTS FOR THE POLY- INTEGER
22 NOMIAL IN X3 DIRECTION
23 IPRINT CONTROL QUANTITY OF PRINT OUT INTEGER
24 *2 PRINT EQUATION SYSTEM BEFORE
25 AND AFTER THE GAUSS JORDAN
26 ELIMINATION
27 *1 PRINT EQUATION SYSTEM AFTER
28 THE GAUSS JORDAN ELIMINATION.
29 *0 PRINT BASIC EQUATIONS WITH
30 ACTUAL PARAMETERS.
31 *0 NO OUTPUT AT ALL.
32 ISTAT CONTROL STATISTICS TO BE PRINTED INTEGER
33 /ISTAT/=2 PRINT COORDINATES OF DATA
34 AND APPROXIMATED POINTS
35 AND DEVIATIONS (ERROR).
36 /ISTAT/=1 ONLY PRINT ERROR.
37 ISTAT/=0 GET ABSOLUTE ERROR.
38 ISTAT/=0 GET RELATIVE ERROR.
39 SUBM1(M,M) SUBMATRIX 1. CLASS IN THE SYSTEM REAL
40 MATRIX.
41 M DIMENSIONING VARIABLE FOR SUBM1 *M*M INTEGER
42 SUBM2(N,N) SUBMATRIX 2. CLASS IN THE SUBMATRIX REAL
43 1. CLASS IN THE SYSTEM
44 ERROR FLAG RETURNED TO THE CALLING INTEGER
45 PROGRAM.
46 *1 NO ERROR DETECTED. GOOD JOB.
47 *1 AN ERROR OCCURRED IN THE GAUSSIAN
48 ELIMINATION. COEFFICIENTS ARE NOT
49 ACCURATE. REVIEW PARAMETERS.
50 .....
51 X1(9,7,5) X1-COORDINATES OF THE DATA POINTS REAL
52 X2(9,7,5) X2-COORDINATES OF THE DATA POINTS REAL
53 X3(9,7,5) X3-COORDINATES OF THE DATA POINTS REAL
54 Y(9,7,5) Y-COORDINATES OF THE DATA POINTS REAL
55 A(64,64) SYSTEM MATRIX. REAL
56 B(64) RIGHT HAND SIDE VECTOR. REAL
57 .....
58 IMPORTANT: THE DATA POINTS X1, X2, X3 AND Y HAVE TO BE PRO-
59 VIDED THROUGH THE LABELED COMMON BLOCK DATA AND
60 THE SYSTEM MATRIX A AND THE RIGHT HAND SIDE VECTOR
61 B ARE AVAILABLE THROUGH THE LABELED COMMON BLOCK
62 DATA. THIS SAVES A LOT OF THAT PRECIOUS THING
63 CALLED MEMORY.
64 .....
65 COMMON / DATA4 / X1,X2,X3,Y
66 COMMON / A84 / A(64)
67 .....
68 REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)
69 REAL A(64,64),B(64)
70 .....
71 REAL COEF(64),SUBM1(M,M),SUBM2(N,N),STAT(9,7,5)
72 .....
73 FORMATS MAT4 START
74 101 FORMAT (///7M AFTER 13,4M STEPS NO SOLUTION WITHIN THE ACCURACY
75 OF 1E-10.3,11M ) (//11M)
76 501 FORMAT (///31M ORDER OF APPROXIMATION POLYNOMIAL IN X1 DIRECTION,
77 52M ORDER OF APPROXIMATION POLYNOMIAL IN X2 DIRECTION,52M,13
78 53M ORDER OF APPROXIMATION POLYNOMIAL IN X3 DIRECTION,52M,13
79 54M NUMBER OF DATA POINTS WITH CONSTANT X1,15M,13/42M NUMBER OF
80 55M DATA POINTS WITH CONSTANT X2,15M,13/42M NUMBER OF DATA POINTS WITH
81 56M CONSTANT X3,15M,13/42M)
82 602 FORMAT (///62M SYSTEM MATRIX A AND VECTOR B BEFORE GAUSS JORDAN E

```

```

      *ELIMINATION/
      900 FORMAT (14X,13(8X,12)))
      901 FORMAT (12X,12,13(14X,1PE9,2),1X/1(4X,3(17,1PE9,2)))
      902 FORMAT (11/140M EQUATION SYSTEM AFTER JORDAN ELIMINATION/
      903 FORMAT (11/120M COEFFICIENTS COEF(I,J,K)/)
      904 FORMAT (2X,3M1 =I2)
      905 FORMAT (12X,3M1 =I2,11(8X,12),1X/3(7X,12(8X,12/))
      906 FORMAT (2X,3M1 =I2,12(12(16,5,3),1X/3(7X,12(11X,5,3/))
      907 FORMAT (11/139M COORDINATES JP DATA POINTS = AS GIVEN)
      908 FORMAT (18M Y(I,J),12,14))
      909 FORMAT (11/140M COORDINATES JP DATA POINTS = AS APPROXIMATED)
      910 FORMAT (11/130M ABSOLUTE ERROR AT EACH POINT)
      911 FORMAT (11/13M ERROR (I,J),12,14))
      912 FORMAT (11/13M RELATIVE ERROR AT EACH POINT IN %)
      913 FORMAT (11/12M STATISTICS SUMMARY//7X,13(AVERAGE ERROR,254,1PE9,
      914 *27X,19MAXIMUM ERROR AT I=I3,3X,2HJ=J3,2X,2HJ=J3,1X,1PE9,2)
      915 FORMAT (11/12M STATISTICS SUMMARY//7X,13(AVERAGE ERROR,254,2PE9,
      916 *27X,19MAXIMUM ERROR AT I=I3,3X,2HJ=J3,2X,2HJ=J3,1X,1PE9,2)
      917 FORMAT (11/17M AFTER 13,14 STEPS NO SOLUTION WITHIN THE ACCURACY
      918 *OF 10.3,11M : I QUIT./)
      919 FORMAT (MATZ STOP)
      C
      LI=LOG(LI/ISSN)
      LO=0
      IERROR=1
      ITRM=50
      EPSLON=.100E-07
      115 PRINT APPROXIMATION PARAMETERS, IF IPRINT IS GREATER THAN J.
      120 IF (IPRINT.LT.0) GO TO 01
      WRITE (LO, 601) NORDR1,NORDR2,NORDR3,NPNTS1,NPNTS2,NPNTS3
      125 START TO PRESET EDGE SECTION SUBMATRICES 1. CLASS.
      01 IPDR1=0
      LI=2*LI-1
      MN=MN
      DO 02 I=1,LI
      IF (K.GT.LI) GO TO 02
      130 TOP EDGE SECTION SUBMATRICES 1. CLASS.
      ISTART=1
      ISTOP=MN
      JSTART=(K-1)*MN+1
      JSTOP=K*MN
      GO TO 03
      140 RIGHT HAND EDGE SECTION SUBMATRICES 1. CLASS.
      02 ISTART=(K-1)*MN+1
      ISTOP=(K-1)*MN+M
      JSTART=(LI-1)*MN+1
      JSTOP=L*MN
      145 PRESET EDGE SECTION SUBMATRICES 1. CLASS IN SYSTEM MATRIX USING 7AT41.
      03 CALL MAT41 (NPNTS1,NPNTS2,NPNTS3,M,N,IPDR1,SUBM1,MN,SUBM2)
      LI=0
      DO 04 I=ISTART,ISTOP,1
      LI=LI+1
      JI=0
      DO 05 J=JSTART,JSTOP,1
      JI=JI+1
      155 AT(J)=SUBM1(I,JI)
      05 IPDR1=IPDR1+1
      160 COPY DEFINED SUBMATRICES 1. CLASS DIAGONALLY.
      LI=LI-1
      DO 07 I=1,LI,1
      DO 07 J=1,LI,1
      ISTART=(I-1)*MN+1
      ISTOP=I*MN
      JSTART=(J-1)*MN+1
      JSTOP=J*MN
      165 DO 06 I=ISTART,ISTOP,1
      DO 06 J=JSTART,JSTOP,1
      170 AT(I*MN+J-MN)=A(I,J)
      07 CONTINUE

```

```

      PRESET RIGHT HAND SIDE VECTOR B.
      I=0
175      DO 08 L1=1,L,1
          IPQWR1=L1-1
          DO 09 M1=1,M,1
              IPQWR2=M1-1
180      DO 08 N1=1,N,1
          IPQWR3=N1-1
          I=I+1
          B(I)=S4(NPNTS1,NPNTS2,NPNTS3,IPQWR1,IPQWR2,IPQWR3,1)
          N1=L*M*N
      PRINT SYSTEM MATRIX AND RIGHT HAND SIDE VECTOR B, IF IPRINT
      IS SET TO 2.
      IF (IPRINT.NE.2) GO TO 10
      WRITE (LO, 602)
190      WRITE (LO, 603) (J,J=1,N1,1)
      DO 09 I=1,N1,1
195      09 WRITE (LO, 604) I,(A(I,J),J=1,N1,1),B(I)
      WRITE (LO, 603) (J,J=1,N1,1)
      IPRINT=1
      GAUSS JORDAN ELIMINATION.
200      10 EPSLON=EPSLON*N1
      CALL ISL4(N1,ITER,ITMAX,EPSON)
      IF (ITER.LT.C) GO TO 32
      PRINT EQUATION SYSTEM, IF IPRINT IS SET TO 1.
      IF (IPRINT.NE.1) GO TO 12
205      WRITE (LO, 605)
      WRITE (LO, 603) (J,J=1,N1,1)
      DO 11 I=1,N1,1
210      11 WRITE (LO, 604) I,(A(I,J),J=1,N1,1),B(I)
      WRITE (LO, 603) (J,J=1,N1,1)
      DEFINE COEFFICIENTS AND PRINT THEM, IF IPRINT IS GREATER THAN 3.
215      12 CONTINUE
      DO 13 I=1,N1,1
          COE(I)=B(I)
220      13 IF (IPRINT.LT.3) GO TO 16
      WRITE (LO, 606)
      DO 13 I=1,N1,1
          WRITE (LO, 607) I
          WRITE (LO, 608) (I3,I3=1,4,1)
225      DO 14 I2=1,M,1
          ICSTRT=((I-1)*N+(I2-1)*N+1)
          ICSTOP=((I-1)*N+I2*N)
          14 WRITE (LO, 609) I2,(C(I,I2),I=ICSTRT,ICSTOP,1)
          15 WRITE (LO, 608) (I3,I3=1,N,1)
      STATISTICS, IF ISTAT ANYTHING ELSE BUT 0.
230      16 IF (ISTAT.EQ.0) GO TO 34
      IF (ISTAT.LT.0) ERROR=1
      IF (ISTAT.GT.0) ERROR=2
      ISTAT=ABS(ISTAT)
      COMPUTE ABSOLUTE ERROR.
235      [=(APPROXIMATED VALUE - GIVEN VALUE) ]
      DO 17 I1=1,NPNTS1,1
          DO 17 I2=1,NPNTS2,1
          DO 17 I3=1,NPNTS3,1
240      17 STAT(I1,I2,I3)=4*(C(I1,I2,I3)*X2(I2,I2,I3)+X3(I1,I2,I3)*NORO
          *R1,NORO2,NORO3)-Y(I1,I2,I3)
      COMPUTE RELATIVE ERROR, IF ISTAT WAS LESS THAN ZERO, I.E. ERROR=1.
245      [=(APPROXIMATED VALUE - GIVEN VALUE)/(GIVEN VALUE) ]
      IF (IERROR.NE.1) GO TO 19
      DO 18 I1=1,NPNTS1,1
          DO 18 I2=1,NPNTS2,1
          DO 18 I3=1,NPNTS3,1
250      18 STAT(I1,I2,I3)=STAT(I1,I2,I3)/Y(I1,I2,I3)
      ERROR STATISTICS: COMPUTE AVERAGE ERROR AND FIND LOCATION
      OF MAXIMUM ERROR.
255

```

```

19 ERRAVE=.0
ERRMAX=ABS(STAT(1,1,1))
IMAX=1
JMAX=1
KMAX=1
DO 20 I1=1,NPNTS1,1
DO 20 I2=1,NPNTS2,1
DO 20 I3=1,NPNTS3,1
IF (ABS(STAT(I1,I2,I3)).LE.ERRMAX) GO TO 20
ERRMAX=STAT(I1,I2,I3)
IMAX=I1
JMAX=I2
KMAX=I3
20 ERRAVE=ERRAVE+ABS(STAT(I1,I2,I3))
ERRAVE=ERRAVE/(NPNTS1*NPNTS2*NPNTS3)
IF (ISTAT.NE.2) GO TO 25

PRINT ORDINATES OF DATA POINTS AND APPROXIMATED VALUES,
IF /ISTAT/ IS EQUAL 2.

WRITE (LO, 510)
DO 22 I3=1,NPNTS3,1
WRITE (LO, 511) I3
WRITE (LO, 512) I2=1,NPNTS2,1
DO 22 I2=1,NPNTS2,1
21 WRITE (LO, 513) I1=1,NPNTS1,1
22 WRITE (LO, 514) I1,I2,I3,STAT(I1,I2,I3)
WRITE (LO, 515) I1,I2,I3,ERRAVE
DO 23 I3=1,NPNTS3,1
WRITE (LO, 516) I3
WRITE (LO, 517) I2=1,NPNTS2,1
DO 23 I2=1,NPNTS2,1
WRITE (LO, 518) I1=1,NPNTS1,1
23 WRITE (LO, 519) I1,I2,I3,STAT(I1,I2,I3),X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3)
24 WRITE (LO, 520) I1,I2,I3,STAT(I1,I2,I3),X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3)
ISTAT=1

PRINT ERRORS, IF /ISTAT/ WAS 1 OR 2.
25 IF (ISTAT.NE.1) GO TO 31
IF (ITEROR.EQ.1) GO TO 28
PRINT ABSOLUTE ERRORS.
WRITE (LO, 521)
DO 27 I3=1,NPNTS3,1
WRITE (LO, 522) I3
WRITE (LO, 523) I2=1,NPNTS2,1
DO 27 I2=1,NPNTS2,1
26 WRITE (LO, 524) I1=1,NPNTS1,1
27 WRITE (LO, 525) I1,I2,I3,STAT(I1,I2,I3),X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3)
GO TO 31

PRINT RELATIVE ERRORS.
28 WRITE (LO, 526)
DO 30 I3=1,NPNTS3,1
WRITE (LO, 527) I3
WRITE (LO, 528) I2=1,NPNTS2,1
DO 30 I2=1,NPNTS2,1
29 WRITE (LO, 529) I1=1,NPNTS1,1
30 WRITE (LO, 530) I1,I2,I3,STAT(I1,I2,I3),X1(I1,I2,I3),X2(I1,I2,I3),X3(I1,I2,I3)
GO TO 31

PRINT ERROR STATISTICS.
31 IF (ITEROR.EQ.2) WRITE (LO, 531) ERRAVE,IMAX,JMAX,KMAX,ERRMAX
IF (ITEROR.EQ.1) WRITE (LO, 532) ERRAVE,IMAX,JMAX,KMAX,ERRMAX
GO TO 34

FATAL ERROR DETECTED. OUTPUT ERROR MESSAGE AND PRESET COEFFICIENTS.
32 ITER=ITER
ITEROR=ITEROR
3(1)=Y(1,1,1)
DO 33 I1=1,NPNTS1,1
DO 33 I2=1,NPNTS2,1
33 WRITE (LI, 101) ITER,EPSLOW
WRITE (LO, 533) ITER,EPSLOW
GO TO 12

RETURN.
34 RETURN
END

```

```

1 SUBROUTINE MAT41 (NPNTS1,NPNTS2,NPNTS3,M,N,IPOWR1,SUBM1,M1,SUBM2)
2 .....
3 PRESET SUBMATRIX 1. CLASS SUBM1 IN THE SYSTEM MATRIX A FOR A
4 40-POLYNOMIAL APPROXIMATION.
5 .....
6 AUTHOR: HANS ZERNER
7 DATE: DECEMBER 31, 1980
8 .....
9 VARIABLE EXPLANATION TYPE
10 .....
11 NPNTS1 NUMBER OF DATA POINTS AT CONSTANT 11 INTEGER
12 NPNTS2 NUMBER OF DATA POINTS AT CONSTANT 12 INTEGER
13 NPNTS3 NUMBER OF DATA POINTS AT CONSTANT 13 INTEGER
14 M NUMBER OF COEFFICIENTS FOR THE POLY-
15 NOMIAL IN X2 DIRECTION. INTEGER
16 N NUMBER OF COEFFICIENTS FOR THE POLY-
17 NOMIAL IN X3 DIRECTION. INTEGER
18 IPOWR1 POWER FOR THE X1 VALUES IN THE SUMMATION. INTEGER
19 .....
20 SUBM1(M,M) SUBMATRIX 1. CLASS IN THE SYSTEM REAL
21 M1 DIMENSIONING VARIABLE FOR SUBM1-M-MOM INTEGER
22 SUBM2(N,N) SUBMATRIX 2. CLASS IN THE SUBMATRIX REAL
23 1. CLASS.
24 .....
25 REAL SUBM1(M,M),SUBM2(N,N)
26 IPWR2=0
27 M1=M-1
28 DO 04 K=1,M1
29 IF (K.GT.M) GO TO 01
30 .....
31 TOP EDGE SECTION SUBMATRICES 2. CLASS.
32 .....
33 ISTART=1
34 ISTOP=M
35 JSTART=(K-1)*N+1
36 JSTOP=M*N
37 GO TO 02
38 .....
39 RIGHT HAND EDGE SECTION SUBMATRICES 2. CLASS.
40 .....
41 01 ISTART=(K-M1)*N+1
42 ISTOP=(K+1)*N+1
43 JSTART=(M-1)*N+1
44 JSTOP=M*N
45 .....
46 PRESET EDGE SECTION SUBMATRICES 2. CLASS IN SUBMATRIX 1. CLASS
47 USING MAT42.
48 .....
49 02 CALL MAT42 (NPNTS1,NPNTS2,NPNTS3,M,IPWR1,IPWR2,SUBM2)
50 .....
51 DO 03 I=ISTART,ISTOP,1
52 IF (I=1)
53 IF (I=1)
54 DO 03 J=JSTART,JSTOP,1
55 IF (J=1)
56 03 SUBM1(I,J)=SUBM2(I,J)
57 04 IPWR2=IPWR2+1
58 .....
59 COPY DEFINED SUBMATRICES 2. CLASS DIAGONALLY.
60 .....
61 M1=M-1
62 DO 06 I1=1,M1,1
63 DO 06 I2=2,M1,1
64 ISTART=(I1-1)*N+1
65 ISTOP=I1*N
66 JSTART=(I2-1)*N+1
67 JSTOP=I2*N
68 DO 05 I=ISTART,ISTOP,1
69 DO 05 J=JSTART,JSTOP,1
70 05 SUBM1(I,N,J-M)=SUBM1(I,J)
71 06 CONTINUE
72 RETURN
73 END

```

SUBROUTINE MAT42 (NPNTS1,NPNTS2,NPNTS3,NB,IPOWR1,IPOWR2,SUBM2)

.....
 PRESET SUBMATRIX 2, CLASS SUBM2 IN THE SUBMATRIX 1, CLASS
 SUBM1 FOR A 40-POLYNOMIAL APPROXIMATION.

AUTHOR: JANS ZERNER
 DATE: DECEMBER 31, 1980

VARIABLE	EXPLANATION	TYPE
NPNTS1	NUMBER OF DATA POINTS AT CONSTANT X1	INTEGER
NPNTS2	NUMBER OF DATA POINTS AT CONSTANT X2	INTEGER
NPNTS3	NUMBER OF DATA POINTS AT CONSTANT X3	INTEGER
N	NUMBER OF COEFFICIENTS FOR THE POLYNOMIAL IN X3 DIRECTION	INTEGER
IPOWR1	POWER FOR THE X1 VALUES IN THE SUMMATION	INTEGER
IPOWR2	POWER FOR THE X2 VALUES IN THE SUMMATION	INTEGER
SUBM2(N,N)	SUBMATRIX 2, CLASS IN THE SUBMATRIX 1, CLASS	REAL

.....
 REAL SUBM2(N,N)

IPOWR3=0
 N1=2*N-1
 DO 03 X=1,N1,1
 IF (K.EQ.N) GO TO 01

TOP EDGE SECTION ELEMENTS

I=1
 J=K
 GO TO 02

RIGHT HAND EDGE SECTION ELEMENTS.

01 I=(K-N)+1
 J=N

PRESET SECTION ELEMENTS IN SUBMATRIX 2, CLASS USING 54.

02 SUBM2(I,J)=54*(NPNTS1,NPNTS2,NPNTS3,IPOWR1,IPOWR2,IPOWR3,J)

03 IPOWR3=IPOWR3+1

N1=N-1

00 04 I=1,N1,1

00 04 J=2,N+1

04 SUBM2(I,J)=SUBM2(I,J)

RETURN

END

REAL FUNCTION S4 (NPNTS1,NPNTS2,NPNTS3,IPWR1,IPWR2,IPWR3,LY)

PERFORM SUMMATIONS TO PRESET THE ELEMENTS A(I,J) IN THE
SYSTEM MATRIX A AND THE ELEMENTS S(I) IN THE RIGHT HAND
SIDE VECTOR S FOR THE LINEAR EQUATION SYSTEM THAT MUST
BE SOLVED FOR A POLYNOMIAL 40-APPROXIMATION.

AUTHOR: HANS ZESNER
DATE: DECEMBER 31, 1980

VARIABLE	EXPLANATION	TYPE
NPNTS1	NUMBER OF DATA POINTS AT CONSTANT X1	INTEGER
NPNTS2	NUMBER OF DATA POINTS AT CONSTANT X2	INTEGER
NPNTS3	NUMBER OF DATA POINTS AT CONSTANT X3	INTEGER
IPWR1	POWER FOR THE X1 VALUES IN THE SUMMATION.	INTEGER
IPWR2	POWER FOR THE X2 VALUES IN THE SUMMATION.	INTEGER
IPWR3	POWER FOR THE X3 VALUES IN THE SUMMATION.	INTEGER
LY	CONTROL PARAMETER =0 Y VALUES ARE NOT INCLUDED IN THE SUMMATION. =1 Y VALUES ARE INCLUDED IN THE SUMMATION. (= RIGHT HAND SIDE VECTOR ELEM.)	INTEGER
S4	RESULT OF SUMMATION.	REAL

X1(9,7,5) X1-COORDINATES OF THE DATA POINTS REAL
X2(9,7,5) X2-COORDINATES OF THE DATA POINTS REAL
X3(9,7,5) X3-COORDINATES OF THE DATA POINTS REAL
Y(9,7,5) Y-COORDINATES OF THE DATA POINTS REAL

IMPORTANT: THE DATA POINTS X1, X2, X3 AND Y HAVE TO BE PROVIDED THROUGH THE LABELED COMMON BLOCK DATA.

COMMON / DATA4 / X1,X2,X3,Y
REAL X1(9,7,5),X2(9,7,5),X3(9,7,5),Y(9,7,5)

S=0
DO 07 K1=1,NPNTS1,1
DO 07 K2=1,NPNTS2,1
DO 07 K3=1,NPNTS3,1
IF (IPWR1.EQ.0) GO TO 01
S1=X1(K1,X2,K3)**IPWR1
GO TO 02

01 S1=1
02 IF (IPWR2.EQ.0) GO TO 03
S2=X1(K1,X2,K3)**IPWR2
GO TO 04

03 S2=1
04 IF (IPWR3.EQ.0) GO TO 05
S3=X1(K1,X2,K3)**IPWR3
GO TO 06

05 S3=1
06 Y=X1(K1,X2,K3)
IF (LY.EQ.0) SY=1.
07 SY=S1*(S2*S3)*SY

S4=SY
RETURN
END

1 INTEGER FUNCTION IEL44 (N)

.....
 * GAUSS JORDAN ELIMINATION WITH MATRIX A AND VECTOR B. THE
 * VARIABLE IEL44 IS SET TO 1, IF THE ELIMINATION IS COMPLETED.
 * IEL44 LESS THAN 0 INDICATES AN ERROR.

* AUTHOR: HANS ZEBNER
 * DATE: DECEMBER 31, 1980

VARIABLE	EXPLANATION	TYPE
N	NUMBER OF EQUATIONS	INTEGER
IEL44	CONTROL PARAMETER, RETURNED TO THE CALLING ROUTINE	INTEGER

* 1 GAUSS JORDAN ELIMINATION WAS PERFORMED CORRECTLY.
 * 0 ERROR ENCOUNTERED DURING GAUSS JORDAN ELIMINATION.

* A(64,64) SYSTEM MATRIX A REAL
 * B(64) RIGHT HAND SIDE VECTOR B REAL

* IMPORTANT: MATRIX A AND VECTOR B HAVE TO BE PROVIDED THROUGH THE LABELED COMMON BLOCK #14. BOTH A AND B WILL UNDERGO CHANGES DURING EXECUTION. I.e. THE ELIMINATION.

.....
 COMMON / A14 / A,B

REAL A(64,64),B(64)

```

35 IEL44=1
   IEL44=1
   EPS=1.00E-030
   DO 06 J=1,N+1
   DO 02 K=1,N+1
40 IF (K.EQ.1) GO TO 02
   CONST=A(K,1)/A(1,1)
   DO 01 J=1,N+1
   A(K,J)=A(K,J)-CONST*A(1,J)
45 IF (J.EQ.1) A(K,J)=0
01 CONTINUE
   B(K)=B(K)-CONST*B(1)
02 CONTINUE
   CONST=A(1,1)
   DO 03 J=1,N+1
50 A(1,J)=A(1,J)/CONST
   B(1)=B(1)/CONST
   DO 06 J=1,N+1
55 ERROR=0
   DO 04 K=1,N+1
   IF (ABS(A(K,K)).GT.EPS) GO TO 04
   ERROR=ERROR+1
04 CONTINUE
   IF (ABS(B(1)).GT.EPS) GO TO 05
   ERROR=ERROR+1
60 IF (ERROR.GT.N+1) GO TO 07
06 CONTINUE
   GO TO 08
07 IEL44=-1
65 IEL44=1
   RETURN
   END

```


176

C

```

      FI=0
      DO 66 I=1,M,1
      90 06 G(I)=0
      DO 68 J=1,M,1
      07 08 G(I)=G(I)+V(I,J)**2
      08 10 I=J,M,1
      99 09 J=J-1
      DO 10 J=J,M,1
      SK=0
      DO 09 I3=1,M,1
      09 SK=SK+V(I3,I)*V(I3,I)
      F=SK/(G(I)*G(I))
      IF (ABS(F).GT.ABS(FI)) FI=F
      10 CONTINUE
      FI=ABS(FI)
      FI=ACOS(FI)
      FI=(FI*180.)/PI

```

START ITERATION TO FIND AND IMPROVE SOLUTION.

```

      DO 27 LX=1,999,1
      110 ITER=LX
      DO 13 J=1,M,1

```

PIVOTISATION AND GAUSSIAN STEP FOR COLUMN J.

```

      119 IX=J
      EV=V(J,J)
      DO 12 I=J,M,1
      IF (ABS(V(I,J)).GT.ABS(EV)) GO TO 11
      120 12 GO TO 12

```

```

      11 IX=I
      EV=V(I,J)
      12 CONTINUE
      IF (IX.EQ.J) GO TO 14
      DO 13 I=1,M,1

```

```

      129 G(I)=V(I,IX)
      V(I,IX)=V(I,J)
      13 V(J,I)=G(I)
      G(I)=0
      14 M=M-J+1

```

```

      DO 15 I=1,M,M,1
      WV=V(I,J)/V(J,J)
      DO 15 KK=J,M
      V(I,KK)=V(I,KK)-V(J,KK)*WV
      15 CONTINUE

```

```

      139 IF (V(I,I).EQ.0) GO TO 32
      SOLUTION.

```

```

      DO 16 I=1,M,1
      140 16 G(I)=0
      G(I)=V(N,M)/V(N,M)
      DO 19 I=1,M,1
      IF (V(I,I).EQ.0) GO TO 32
      149 19 G(I)=V(I,M)
      DO 17 J=1,M,1
      JM=J
      IF (J.EQ.I) GO TO 19
      150 17 G(I)=G(I)-G(J)*V(I,J)
      18 CONTINUE
      19 G(I)=G(I)/V(I,I)

```

CORRECT SOLUTION BY AMOUNT OF ERROR WHICH WAS INVOLVED IN THIS ITERATION STEP. TAKE THIS VECTOR AS START VECTOR FOR THE NEXT ITERATION.

```

      DO 20 I=1,M,1
      160 20 G(I)=8(I)+G(I)

```

COMPUTE THE REMAINING ERROR.

```

      DO 22 I=1,M,1
      DO 21 J=1,M,1
      165 21 V(I,M)=V(I,M)+G(I)*A(I,J)
      22 V(I,M)=A(I,M)-V(I,M)
      R=0
      DO 23 I=1,M,1
      170 23 R=R+V(I,M)*V(I,M)

```

```

      R=SQRT(1)
      IF (R.LT.EPSLOM) GO TO 28
      IF (LA.GT.ITRMAX) GO TO 33
      DO 24 I=1,N+1
175      24 G(I)=G(I)
      PRINT SYSTEM MATRIX A AND RIGHT HAND SIDE VECTOR B AFTER THE
      PIVOTISATION.
180      IF (IPRINT) WRITE (LJ, 504) LX
      IF (IPRINT) WRITE (LJ, 502) (JJ=1,N+1)
      DO 25 I=1,N+1
185      25 IF (IPRINT) WRITE (LJ, 503) I,(V(I,J),J=1,N+1),G(I)
      IF (IPRINT) WRITE (LJ, 502) (JJ=1,N+1)
      DO 26 I=1,N+1
      DO 26 J=1,N+1
      26 V(I,J)=A(I,J)
      STOP LOOP TO FIND SOLUTION.
190      27 CONTINUE
      REARRANGE SOLUTION IN THE ARRAY V.
195      28 CONTINUE
      DO 29 I=1,N+1
      29 V(I,65)=G(I)
      PRINT SOLUTION, IF IPRINT IS SET .TRUE. .
200      30 IF (IPRINT) WRITE (LJ, 505) 4
      IF (IPRINT) WRITE (LJ, 505) (I,G(I),I=1,N+1)
      IF (IPRINT) WRITE (LJ, 507) F1
      IF (IPRINT) WRITE (LJ, 508) 2,LX
205      DRUCKEN DER MATRIX A UND DES VECTORS B NACH ELIMINATION.
      IF (IPRINT) WRITE (LJ, 613)
      IF (IPRINT) WRITE (LJ, 602) (JJ=1,N+1)
210      DO 31 I=1,N+1
      31 IF (IPRINT) WRITE (LJ, 603) I,(V(I,J),J=1,N+1)
      IF (IPRINT) WRITE (LJ, 602) (JJ=1,N+1)
      RETURN
215      ERROR ENCOUNTERED. STOP.
      32 ITER=1
      WRITE (LJ, 509) I,I=V(I,I)
      G(I)=G(I)/V(I,I)
      RETURN
220      NO CONVERGENCE OF THE SOLUTION.
      33 ITER=ITER
      RETURN
225      END

```

```

1      REAL FUNCTION F4 (CDEF,X1,X2,X3,NOROR1,NOROR2,NOROR3)
2      .....
3      * COMPUTE FUNCTION VALUE Y=F(X1,X2,X3) OF A 40-POLYNOMIAL USING
4      * THE HURNER SCHEME IN ORDER TO INCREASE SPEED AND ACCURACY.
5      *
6      * AUTHOR:      HANS ZERNER
7      * DATE:        DECEMBER 31, 1980
8      *
9      * VARIABLE    EXPLANATION                                TYPE
10     * CDEF(104)   POLYNOMIAL COEFFICIENTS                    REAL
11     * X1           X1-VALUE, WHERE Y WILL BE COMPUTED.         REAL
12     * X2           X2-VALUE, WHERE Y WILL BE COMPUTED.         REAL
13     * X3           X3-VALUE, WHERE Y WILL BE COMPUTED.         REAL
14     * NOROR1       ORDER OF THE POLYNOMIAL IN THE              INTEGER
15     *               X1 DIRECTION.
16     * NOROR2       ORDER OF THE POLYNOMIAL IN THE              INTEGER
17     *               X2 DIRECTION.
18     * NOROR3       ORDER OF THE POLYNOMIAL IN THE              INTEGER
19     *               X3 DIRECTION.
20     * F4           ORDINATE OF THE 40-POLYNOMIAL.              REAL
21     *
22     * .....
23     REAL CDEF(104),D(54),E(54)
24     NCDEF1=NOROR1+1
25     NCDEF2=NOROR2+1
26     NCDEF3=NOROR3+1
27
28     * COMPUTE COEFFICIENTS D BASED ON COEFFICIENTS CDEF AND X3
29     *
30     DO 02 I1=1,NCDEF1,1
31     DO 02 I2=1,NCDEF2,1
32     IC=(I1-1)*NCDEF2+I2*NCDEF3
33     D=(I1-1)*NCDEF2+I2
34     Y=CDEF(IC)
35     DO 04 I3=1,NOROR3,1
36     I3=NCDEF3-I3+1
37     IC=(I1-1)*NCDEF2+I2*NCDEF3+I3
38     Y=CDEF(IC)+I3*Y
39     02 D(I1)=Y
40
41     * COMPUTE COEFFICIENTS E BASED ON COEFFICIENTS D AND X2.
42     *
43     DO 04 I1=1,NCDEF1,1
44     DO 04 I2=1,NCDEF2,1
45     Y=D(I1)
46     DO 03 I3=1,NOROR2,1
47     I3=NCDEF2-I3+1
48     Y=(I3-1)*NCDEF2+I2
49     Y=D(I1)+I3*Y
50     03 E(I1)=Y
51
52     * COMPUTE Y BASED ON COEFFICIENTS E AND X1.
53     *
54     Y=E(INCDEF1)
55     DO 05 I1=1,NOROR1,1
56     I1=NCDEF1-I1+1
57     Y=E(I1)+I1*Y
58     05 Y=Y
59     RETURN
60     END

```

DISTRIBUTION LIST

	<u>No. of Copies</u>
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	1
2. Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Office of Research Administration Code 012A Naval Postgraduate School Monterey, California 93940	1
4. Chairman Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5. Director, Turbopropulsion Laboratory Department of Aeronautics Naval Postgraduate School Monterey, California 93940	30
6. Dr. Gerhard Heiche Naval Air Systems Command Code AIR-310 Navy Department Washington, D.C. 20360	1
7. Mr. Karl H. Guttman Naval Air Systems Command Code AIR-330 Navy Department Washington, D.C. 20360	1
8. Mr. James R. Patton, Jr. Power Program, Code 473 Office of Naval Research Arlington, Virginia 22218	1
9. Dr. A. D. Wood Office of Naval Research Eastern/ Central Regional Office 666 Summer Street Boston, Massachusetts 02210	1

10. Commanding Officer 1
Naval Air Propulsion Test Center
Attn: Mr. Vernon Lubosky
Trenton, New Jersey 08628
11. National Aeronautics & Space Administration 1
Lewis Research Center (Library)
2100 Brookpark Road
Cleveland, Ohio 44135
12. CAG Library 1
The Boeing Company
Seattle, Washington 98124
13. Library 1
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215
14. Library 1
Pratt and Whitney Aircraft
Post Office Box 2691
West Palm Beach, Florida 33402
15. Library 1
Pratt and Whitney Aircraft
East Hartford, Connecticut 06108
16. Chief, Fan and Compressor Branch 1
Mail Stop 5-9
NASA Lewis Research Center
2100 Brookpark Road
Cleveland, Ohio 44135
17. Prof. D. Adler 1
Technion Israel Institute of Technology
Department of Mechanical Engineering
Haifa 32000
Israel
18. Director, Whittle Laboratory 1
Department of Engineering
Cambridge University
England

19. Prof. F. A. E. Breugelmans 1
Institut von Karman de la Dynamique
des Fluides
72 Chausee de Waterloo
1640 Rhode-St. Genese
Belgium
20. Library 1
Air Research Mfg. Corporation
Division of Garrett Corporation
402 South 36th Street
Phoenix, Arizona 85034
21. Dr. Robert P. Dring 1
United Technologies Research Labs
400 Main Street
Hartford, Connecticut 06108
22. Mr. Jean Fabri 1
ONERA
29, Ave. de la Division Leclerc
92 Chatillon
France
23. Prof. Dr. Ing Heinz E. Gallus 1
Lehrstuhl und Institut fur Strahlantiebe
und Turboarbeitsmaschinen
Rhein.-Westf. Techn. Hochschule Aachen
Templergraben 55
5100 Aachen, Germany
24. Mr. R. A. Langworthy 1
Army Aviation Material Laboratories
Department of the Army
Fort Eustis, Virginia 23604
25. Dr. W. Schlachter 1
Brown, Boveri & Company Ltd.
Department T-T
Haselstrasse
CH-5401 Baden, Switzerland
26. Prof. T. H. Okiishi 1
Professor of Mechanical Engineering
208 Mechanical Engineering Building
Iowa State University
Ames, Iowa 50011

27. Dr. Leroy H. Smith, Jr. 1
Manager, Compressor & Fan Technology Operation
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215
28. Dr. Arthur J. Wennerstrom 1
ARL/LF
Wright-Patterson AFB
Dayton, Ohio 45433
29. Dr. H. Weyer 1
DFVLR
Linder Hohe
505 Porz-Wahn
Germany
30. Mr. P. F. Yaggy 1
Director
U.S. Army Aeronautical Research Laboratory
AMES Research Center
Moffett Field, California 94035
31. Prof. C. H. Wu 1
P.O. Box 2706
Beijing 100080
China
32. Director 1
Gas Turbine Establishment
P.O. Box 305
Jiangyou County
Sichuan Province
China

10 72

11 73

12 74

13 75

14 76

15 77

16 78

17 79

18 80

19 81

20 82

21 83

22 84

23 85

24 86

25 87

26 88

27 89

28 90

29 91

30 92

31 93

32 94

33 95

34 96

35 97

36 98

37 99

38 100

39 101

40 102

41 103

42 104

43 105

44 106

45 107

46 108

47 109

48 110

49 111

50 112

DUDLEY KNOX LIBRARY



3 2768 00347367 9